



# **Tap&Go: Gamificant l'esport en dispositius Android amb NFC**

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes  
realitzat per  
Jonathan Fragoso Pérez  
i dirigit per  
Marta Prim Sabrià

**Escola d'Enginyeria**  
Sabadell, Abril de 2013



La sotasignat, ***Marta Prim Sabrià***,  
professora de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present  
memòria ha estat realitzat sota la seva direcció  
per ***Jonathan Fragoso Pérez***

I per a que consti firma la present.  
Sabadell, ***Abril*** de ***2013***

-----  
Signat: ***Marta Prim Sabrià***





El sotasignat, *Joan Vicent Orenge Serisuelo*, de  
l'empresa *Tempos21 (Atos)*,

**CERTIFICA:**

Que el treball al que correspon la present  
memòria  
ha estat realitzat sota la seva supervisió per

*Jonathan Fragoso Pérez*

I per a que consti firma la present.  
Sabadell, *Abril* de *2013*

-----  
Signat: *Joan Vicent Orenge Serisuelo*



## FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

<b>Títol del projecte:</b> Tap&Go: Gamificant l'esport en dispositius Android amb NFC	
<b>Autor:</b> Jonathan Fragoso Pérez	<b>Data:</b> Abril 2013
<b>Tutora:</b> Marta Prim Sabrià	
<b>Titulació:</b> Enginyeria Tècnica en Informàtica de Sistemes	
<b>Paraules clau</b> <ul style="list-style-type: none"><li>• Català: Aplicació Android, gamificació esport, tecnologia NFC</li><li>• Castellà: Aplicación Android, gamificación deporte, tecnología NFC</li><li>• Anglès: Android application, sports gamification, NFC technology</li></ul>	
<b>Resum del projecte</b> <ul style="list-style-type: none"><li>• Català: El contingut d'aquest projecte, consisteix en desenvolupar una solució per a dispositius mòbils Android amb NFC, amb la finalitat de motivar i incrementar el nombre de ciutadans que practiquen esport a la ciutat de Barcelona. El projecte inclou disseny de la interfície mòbil, la creació d'una base de dades, un servei d'API Rest per accedir a les dades des dels dispositius, l'ús del servei de mapes de Google i diferents llibreries i eines per a dissenyar l'arquitectura del sistema. Es tindran en compte les restriccions en disseny i funcionalitat per part del client (ajuntament).</li><li>• Castellà: El contenido de este proyecto, consiste en desarrollar una solución para dispositivos móviles Android NFC, con la finalidad de motivar e incrementar el número de ciudadanos que practican deporte en la ciudad de Barcelona. El proyecto incluye diseño de la interfície móvil, creación de una base de datos, un servicio de API Rest para acceder a los datos desde los dispositivos, servicio de mapas de Google y diversas librerías y herramientas para diseñar la arquitectura del sistema. Se tendrán en cuenta las restricciones en diseño y funcionalidad del cliente (ayuntamiento).</li></ul>	

- Anglès: The project content consists of developing a solution for all the NFC enabled Android mobile phone devices, looking for a way to motivate and increase the number of citizens which practice sports into the city of Barcelona. The projects features include the UI (user interface) design, a DB (database) creation, a Rest API service to retrieve all the data requested by the devices, the Google Maps service and the usage of different libraries and tools to design the system architecture. It will be also considered the design and features constraints requested by the client (city council).

# Índex

<b>1. Introducció</b>	1-4
1.1- Descripció del projecte	1-2
1.2- Estat de l'art	2
1.3- Motivacions personals	2
1.4- Definicions i acrònims	2-3
1.5- Estructura de la memòria	3-4
<b>2. Estudi de viabilitat</b>	5-28
2.1- Introducció	5-11
2.1.1- Tipologia i keywords (paraules clau)	5
2.1.2- Descripció de la solució	5-6
2.1.3- Objectius del projecte	6-7
2.1.4- Parts interessades (stakeholders)	7-9
2.1.5- Entorn de treball de l'usuari	9-10
2.1.6- Producte i documentació	10
2.1.7- Normativa i legislació	10-11
2.2- Requeriments i restriccions	11-17
2.2.1- Requeriments funcionals	11-15
2.2.2- Requeriments no funcionals	16-17
2.2.3- Restriccions del sistema	17
2.3- Alternatives al mercat i justificació de la solució	17-18
2.4- Planificació	18-24
2.4.1- Descripció de la metodologia emprada: Scrum	18-20
2.4.2- WBS (Work Breakdown Structure)	21
2.4.3- Calendari	22-23
2.4.4- Riscos del projecte	23-24
2.4.4.1- Llistat de riscos i catalogació	
2.4.4.2- Pla de contingència	
2.5- Càlcul de costos (pressupost)	25-26

2.5.1- Costos directes o de personal	25
2.5.2- Costos indirectes: material i llicències	25-26
2.5.3- Resum i anàlisi cost-benefici	26
2.6- Conclusions de viabilitat	27
<b>3. Disseny de l'arquitectura Software: UML</b>	29-32
3.1- Introducció i eines de disseny	29
3.2- Document d'arquitectura	29-31
3.2.1- Diagrama de casos d'ús	29
3.2.2- Diagrama de components	29-30
3.2.3- Diagrama de paquets	31
3.2.4- Diagrama de classes	31
3.3- Disseny de la base de dades	31-32
<b>4. Disseny de la UI (User Interface) mòbil</b>	33-36
4.1- Sketches (primers esbossos)	33
4.2- Coloured wireframes (disseny final)	34-36
<b>5. El servidor i l'API Rest</b>	37-42
5.1- Relació events interfície d'usuari / peticions	37-38
5.2- Implementació del servidor	38-41
<b>6. Desenvolupament de les aplicacions mòbils</b>	43-66
6.1- Restriccions i estructura de codificació	43-46
6.2- El desenvolupament (Sprints)	46-66
6.2.1- Sprint 1	47-51
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
6.2.2- Sprint 2	51-53
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
6.2.3- Sprint 3	53-56
a) Planificació	

b) Implementació	
c) Test i valoració d'Sprint	
6.2.4- Sprint 4	56-58
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
6.2.5- Sprint 5	58-60
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
6.2.6- Sprint 6	60-64
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
6.2.7- Sprint 7	64-66
a) Planificació	
b) Implementació	
c) Test i valoració d'Sprint	
<b>7. Conclusions</b>	67-68
7.1- Assoliment dels objectius	67
7.2- Línies futures	67-68
7.3- Valoració personal	68

## Agraïments

## Bibliografia

## Annex: Diagrames UML i disseny de la UI (User Interface)

## Índex d'il·lustracions

<b>Il·lustració 1:</b> Estructura d'un sprint	21
<b>Il·lustració 2:</b> Diagrama WBS	21
<b>Il·lustració 3:</b> Diagrama de components genèric	30
<b>Il·lustració 4:</b> Diagrama de components del servidor	30
<b>Il·lustració 5:</b> Disseny entitat/relació de la base de dades	32
<b>Il·lustració 6:</b> Taules de la base de dades	32
<b>Il·lustració 7:</b> Icona de l'aplicació	34
<b>Il·lustració 8:</b> Splash activity	34
<b>Il·lustració 9:</b> Rutes	34
<b>Il·lustració 10:</b> Detall de la ruta	34
<b>Il·lustració 11:</b> Detall del punt	35
<b>Il·lustració 12:</b> Històric	35
<b>Il·lustració 13:</b> Premis i reptes	35
<b>Il·lustració 14:</b> Amics	35
<b>Il·lustració 15:</b> Preferències	35
<b>Il·lustració 16:</b> Inici de pràctica	35
<b>Il·lustració 17:</b> Pràctica	36
<b>Il·lustració 18:</b> Resum de pràctica	36
<b>Il·lustració 19:</b> Login admin.	36
<b>Il·lustració 20:</b> Edició d'un punt	36
<b>Il·lustració 21:</b> Diàleg d'afirmació	50
<b>Il·lustració 22:</b> Diàleg cancel·lable	50
<b>Il·lustració 23:</b> Diàleg d'amics	55
<b>Il·lustració 24:</b> Diàleg del calendari	57
<b>Il·lustració 25:</b> Dispositiu de test i etiquetes NFC	63



## Índex de taules

<b>Taula 1:</b> Catalogació de riscos	23
<b>Taula 2:</b> Pla de contingència	24
<b>Taula 3:</b> Recursos directes	25
<b>Taula 4:</b> Recursos indirectes	26
<b>Taula 5:</b> Nomenclatura dels identificadors de cadenes estàtiques	44
<b>Taula 6:</b> Nomenclatura de les imatges	45
<b>Taula 7:</b> Nomenclatura dels layout	46



## 1. Introducció

L'ajuntament de la ciutat de Barcelona vol motivar als ciutadans a fer més esport i que aquests puguin fer-ho de manera senzilla, així com també poder dur a terme un control dels seus avenços. A més a més, com a institució, vol obtenir també dades d'aquesta població, per tal de tenir un control més gran sobre estadístiques de l'edat, pes, alçada, esports més practicats, etc. També voldria controlar les zones, així de manera indirecta addicionalment s'atrau al turisme gaudint dels millors edificis i paisatges dins la ciutat.

Degut a que la ciutat disposa de grans espais oberts com grans parcs, les seves meravelloses platges i zones ja habilitades per a córrer, aquest esport podria ser el punt de partida per a trobar aquest increment d'esportistes. Addicionalment, afegint en aquest grup d'esports el ciclisme, es reactivaria i s'augmentaria el negoci del Bicing, ja que hi ha una gran quantitat d'estacionaments de bicicletes d'aquest servei, distribuïts per tota l'extensió de la ciutat.

El més important doncs, és trobar el mitjà de transmissió per a aconseguir aquests objectius. El mercat de les aplicacions per a telèfons mòbils està revolucionant el món actualment i desenvolupant una solució d'aquest tipus, es podrien obtenir resultats amb un gran criteri d'èxit. Cal afegir que, a més a més, si la solució és òptima, la ciutat incrementarà el seu prestigi de ser una de les ciutats punteres emprant l'ús d'últimes tecnologies mundialment.

### 1.1. Descripció del projecte

Aquestes necessitats i possibilitats, acaben desencadenant i donant forma a aquest projecte: una aplicació mòbil per a dispositius Android, on els usuaris podran dur un control de les seves pràctiques esportives, així com obtenir premis pels seus avenços, i finalment, recomanar les millors rutes i desafiar als seus amics.

Amb la finalitat d'aconseguir un projecte encara més innovador, s'emprarà l'ús de mapes en tres dimensions, el "check-in" (pas pels punts de control) es farà físicament a través dels dispositius mòbils fent ús de la tecnologia NFC (Near Field Communication) i mitjançant unes etiquetes que s'enganxaran en les rutes. L'aplicació s'integrarà amb Facebook, la xarxa social del moment, facilitant així els suggeriments de rutes i els reptes amb els amics que tinguin ambdues aplicacions disponibles al seu mòbil.

El fet d'inserir dades, actualitzar-les, així com assignar les etiquetes corresponents a cadascuna de les rutes, fa que sigui imprescindible una aplicació d'administració que gestionarà l'Ajuntament.

En aquest projecte s'utilitzaran diverses tecnologies com el sistema operatiu Android i NFC, llibreries i APIs (Application programming Interface) per fer ús dels mapes i utilitzar components personalitzats en la interfície d'usuari, així com programari pel desenvolupament del disseny d'aquesta, programar amb Java usant l'SDK (Software Development Kit) d'Android i eines per la fase de disseny de l'arquitectura del sistema i del servidor.

## **1.2. Estat de l'art**

Actualment, no hi ha cap solució que satisfaci les necessitats del client. Únicament hi ha unes rutes, que l'Ajuntament té accessibles al seu web d'esports, però no són tan accessibles per als ciutadans com ho seria una aplicació mòbil on es poden consultar els mapes i s'hi afegeix motivació mitjançant la gamificació a través dels amics i els premis que proporcionarà l'ajuntament als qui aconsegueixen superar els seus reptes.

L'estudiant desenvolupa el projecte a l'empresa on treballa, Tempos 21, un departament dins de la multinacional Atos (multinacional que desenvolupa solucions SW). Aquest departament, s'encarrega de desenvolupar aplicacions per a dispositius mòbils a demanda dels clients, és a dir, un cop el client demana el producte, es pacten els requeriments, funcionalitat i anàlisi i finalment es comença a desenvolupar el producte.

## **1.3. Motivacions personals**

La motivació en aquest projecte es basa en l'aprenentatge de les últimes tecnologies utilitzades en Android, com la gestió i control d'events del sensor HW (Hardware) d'un dispositiu, l'última actualització dels mapes de Google en 3D. A més a més, l'estudiant aprèn a gamificar una aplicació mitjançant una xarxa social, desenvolupar una API Rest per a un servidor, així com obtenir nous coneixements amb arquitectura i disseny de SW, aprenent a optimitzar i a treballar fent un gran nombre d'operacions per tal de fer el "tracking" d'una pràctica esportiva.

De manera indirecta, també aprendrà a utilitzar noves eines per al disseny de la interfície d'usuari i dels diagrames UML, així com a enfrontar-se davant d'un problema real de gran dificultat i solucionar-lo mitjançant una metodologia àgil de gestió de projectes, Scrum, que s'acostuma a utilitzar en equips i per a projectes grans.

## **1.4. Definicions i acrònims**

A continuació s'esmentaran els principals acrònims i abreviacions que s'utilitzaran en

aquest document:

**Pfc:** Projecte de final de carrera. Projecte que realitza un estudiant per finalitzar la seva carrera (aquest projecte forma part dels objectius que ha d'assolir un estudiant abans d'obtenir el títol).

**NFC:** Near Field Communication. És un nou sistema de comunicació que permet la transmissió de dades a molt curta distància, de fet els dos dispositius han d'estar tocant gairebé perquè s'iniciï la comunicació. La velocitat de transmissió no és elevada, per tant s'utilitza més com a un mitjà per iniciar una comunicació de forma ràpida i llavors quan és necessària realment la transmissió de dades, es canvia la comunicació a un altre tipus com per exemple Bluetooth o Wi-fi.

**API:** Application Programming Interface. Interfície d'especificació de codi que permet interactuar entre components SW per facilitar la comunicació entre ells.

**SDK:** Software Development Kit. Conjunt de programari que permet a un desenvolupador de SW programar en una plataforma o sistema operatiu.

**IDE:** Integrated Development Environment. És una aplicació SW que permet desenvolupar i programar codi als enginyers. Normalment conté un editor de codi, un compilador, un entorn d'execució, eines de col·laboració en equip (repositoris de codi) i altres eines addicionals.

**UML:** Unified modeling Language. És una llenguatge de modelat i disseny de SW orientat a objectes. L'UML està format per un estudi, així com un conjunt de diagrames previs al desenvolupament i la codificació, per tal de que els requeriments estiguin ben definits, tenir ben clar l'arquitectura que s'utilitzarà i alhora que tot l'equip pugui entendre de manera senzilla com funciona la solució, així com si en un futur s'han de crear noves actualitzacions sigui fàcil recordar el funcionament i els canvis siguin el més senzills i eficients possibles.

**MySQL:** Gestor de base de dades relacional molt utilitzat en el desenvolupament d'aplicacions que necessiten obtenir, guardar i modificar dades en un servidor.

**UI:** User interface. Quan es faci referència a la UI, s'estarà parlant de la interfície d'usuari mòbil, és a dir de la navegabilitat i el disseny de l'aplicació mòbil que usará l'usuari.

## 1.5. Estructura de la memòria

Un cop s'ha introduït el projecte, es presentarà l'estudi de viabilitat d'aquest. En aquest punt s'exposen els objectius i requeriments de l'aplicació de forma més detallada, s'estudien les diverses alternatives per a obtenir una solució que satisfaci de la forma més òptima possible les necessitats del client. També es comprova que no hi hagi cap

incompliment en la llei de protecció de dades o alguna altra llei, es distribueix l'equip, es tenen en compte els riscos que hi pugui haver i com solucionar-los en cas de que apareguin, es calculen els recursos de SW, HW i de personal necessari. També s'exposarà com es durà a terme la planificació mitjançant Scrum; una metodologia àgil de gestió de projectes que treballa de forma iterativa, dividint el projecte en sprints (com veurem en aquesta part un sprint conté unes tasques que un cop finalitza l'sprint, s'obté una part totalment funcional del projecte).

Finalment, s'elabora un pressupost per concloure si el projecte és viable o no ho és.

Després de l'estudi de viabilitat, comença el disseny de l'arquitectura SW, mitjançant un estudi amb UML, emprant els diagrames necessaris perquè el projecte es desenvolupi satisfactòriament optimitzant el codi i intentant que l'acoblament entre els components sigui mínim, per tal de si s'afegeix funcionalitat en un futur, els canvis a fer siguin mínims.

Aquest apartat de disseny inclou també el disseny de la base de dades on s'emmagatzemaran les dades de les rutes, dels punts de control i dels usuaris per a controlar els temps obtinguts i donar els premis als qui superin els reptes.

Un cop el disseny de l'arquitectura està acabat s'elabora el disseny de la interfície pels dispositius mòbils. El fet de ser una aplicació a demanda, implicarà seguir unes restriccions; es seguirà el propi estil de disseny i colors que proporciona l'ajuntament per a les seves aplicacions.

A continuació, s'esmentarà com s'ha construït el servidor, l'API Rest per accedir als serveis als quals les aplicacions accediran, en resposta als events corresponents en cadascuna de les pantalles (activities en Android).

I un cop el servidor funciona, s'explicarà com s'han implementat les dues aplicacions, detallant cadascun dels sprints en contingut, planificació, implementació a alt nivell, el test i les valoracions obtingudes.

Finalment, es valora si s'han assolit els objectius, així com els problemes que han sorgit, les línies futures i la valoració personal.

## 2. Estudi de viabilitat

### 2.1. Introducció

La finalitat de l'estudi de viabilitat és avaluar les diferents alternatives possibles a l'hora de dur a terme el projecte amb el major grau d'efectivitat.

Per a fer això és necessari un estudi de la viabilitat tècnica, econòmica, legal i operativa. Un cop els requeriments estan ben definits i l'estudi s'ha realitzat correctament, es podrà implementar i desenvolupar el projecte intentant assolir el major criteri d'èxit en el projecte.

#### 2.1.1. Tipologia i keywords (paraules clau)

Aquest projecte es tracta d'un projecte de desenvolupament, ja que es tracta de realitzar una solució software (SW) des de l'inici. Hi haurà dos aplicacions mòbils: una per a poder afegir dades per part d'un administrador, i l'altra permetrà usar l'aplicació als ciutadans de Barcelona, fent ús de les dades que han proporcionat aquests administradors. La plataforma de les aplicacions és Android. S'emprarà la tecnologia NFC (Near field communication), per tant també és necessari hardware (HW) que en aquest cas seran aquestes etiquetes NFC on s'emmagatzemarà la informació dels punts de control que llegiran els dispositius.

Les keywords són aquelles paraules clau que representen i resumeixen la funcionalitat d'un projecte. En aquest cas són:

- \* Aplicació Android amb gamificació
- \* Sport tracking amb NFC

#### 2.1.2. Descripció de la solució

Tap&Go és un nou concepte de xarxa social "física" per a dispositius NFC funcionant amb la plataforma Android, la qual la seva finalitat és motivar al ciutadà a fer esport; en aquest cas, inicialment, es podrà córrer o anar amb bicicleta.

A nivell genèric, es tracta de tenir unes rutes predefinides de les quals l'usuari podrà veure la informació mitjançant la interfície mòbil, per tal de seleccionar el lloc on prefereix anar a practicar l'esport. Durant la pràctica de l'esport, en la ruta, aquest usuari anirà "tocant" (check-in) cadascuna de les etiquetes NFC (punts de control de la ruta) que es va trobant, i es va generant un informe de temps, velocitats, i altres dades que li facilitaran el "tracking" (informació de la ruta).

Els usuaris que es registrin en l'aplicació per tal d'obtenir les funcionalitats de xarxa social i gamificació, facilitaran dades a l'Ajuntament de manera indirecta, un dels objectius del

projecte (aquestes dades ajudaran a l'Ajuntament a fer estudis i estadístiques de la seva població).

D'altra banda, serà necessària una aplicació senzilla d'administració per tal de salvar la informació de cadascun dels punts de control en les etiquetes NFC, així com tirar la fotografia del lloc on és ubicada l'etiqueta perquè els usuaris trobin amb facilitat la localització del punt.

### **2.1.3. Objectius del projecte**

Especificant més detalladament la solució, després d'una sessió de brainstorming (pluja d'idees) amb els comercials, el cap de projectes i l'estudiant que realitza el projecte, s'han definit els següents objectius o requeriments a partir de les necessitats client:

- Per part de l'aplicació dels administradors (Ajuntament de Barcelona):

- 1- Afegir noves rutes amb la seva corresponent informació.
- 2- Afegir punts de control així com informació addicional del lloc.
- 3- Poder associar etiquetes NFC a aquests punts de control per a poder fer el "check-in" físic i associar una foto de la ubicació d'aquestes.

- Per part de l'aplicació dels ciutadans:

Es permetrà l'ús de l'aplicació a través d'un registre amb Facebook per als ciutadans que vulguin fer ús de la gamificació i interacció amb altres usuaris. Els usuaris que no es registrin, podran fer-ne també ús, però no podran utilitzar aquests serveis.

\* La funcionalitat d'un usuari no registrat és la següent:

- 1- Veure unes rutes ja predefinides amb tots els punts on hi ha tags NFC al mapa. L'usuari veurà reflectits en la interfície els llocs per on ja ha passat, i de manera diferent, els punts que li manquen per recórrer. Els punts i les rutes contenen, a més a més, informació sobre el lloc, fotografies, punts d'interès, d'entre altres, i que haurà de poder visualitzar.
- 2- Consultar l'històric de les seves pràctiques i la informació de cadascuna d'aquestes.

\* La funcionalitat addicional d'un usuari registrat és la següent:

- 3- Apartat de fites i premis.
- 4- Poder convidar a superar temps als amics (reptes).
- 5- Compartir rutes, informació de la pràctica i altres, a través de Facebook.
- 6- Consultar la velocitat a la qual va l'esportista en cada moment, així com informació complementària del "tracking" de la pràctica.



7- Cal dur a terme un bon control de les velocitats per tal de que els temps que obtingui un usuari siguin reals en la pràctica de la ruta.

#### **2.1.4. Parts interessades (stakeholders)**

##### **STAKEHOLDERS**

A continuació, es descriuen cadascuna de les parts interessades en el projecte i que per tant hi tenen implicació directa. S'especifica el nom de la persona o entitat interessada i després la seva descripció, la seva responsabilitat en el projecte i el seu criteri d'èxit.

##### **1- Ajuntament de Barcelona**

**Descripció:** Es tracta del client. Hi haurà en coordinador del projecte en nom de l'ajuntament de la ciutat de Barcelona.

**Responsabilitats:** És qui s'encarregarà de parlar i negociar amb el nostre cap de projectes per tal de definir els requeriments, resoldre dubtes, posicionar les fites, donar el vist-i-plau al producte, fixar una data de "release" del producte.

**Criteri d'èxit:** L'aplicació que demana satisfà ben bé les necessitats plantejades, funcionant de manera correcta i seguint un disseny i una implementació en el desenvolupament òptima. Els ciutadans estan contents amb la solució i obté les dades que vol, a més del prestigi com a "ciutat tecnològica".

##### **2- Joan Vicent Orenge Serisuelo**

**Descripció:** És cap de projectes a Tempos21 (departament d'aplicacions mòbils dins de la multinacional Atos).

**Responsabilitats:** S'encarregarà de pactar i negociar amb el client, així com coordinar el projecte perquè tot vagi avançant amb les dates escaients, resoldre dubtes per part del desenvolupador, donar els mitjans necessaris per a dur-se a terme, etc. En definitiva, farà la funció que fa un cap de projectes.

**Criteri d'èxit:** El client està satisfet amb el seu producte i es van entregant les parts corresponents en les dates pactades, gràcies a la bona coordinació amb el desenvolupador (l'estudiant). Un desenvolupador de l'equip adquireix coneixements de NFC i per tant com a coordinador veu les avantatges i problemes d'aquesta tecnologia, que fins ara no s'havia aplicat en altres projectes tot i que s'estava començant a investigar. L'estudiant adquireix nous coneixements intentant obtenir una bona qualificació en el projecte de final de carrera mitjançant la solució aportada.

##### **3- Jorge Salinas Martínez**

**Descripció:** És comercial a Tempos21.

**Responsabilitats:** S'encarregarà de pactar i negociar amb el client, mostrant la demostració d'aquest projecte al client, per tal de poder fer futurs negocis amb aquest client.

**Criteri d'èxit:** L'empresa té guanys econòmics així com de prestigi, el client per tant està content amb el seu producte. S'implementa una primera solució amb NFC a l'oficina, per tant s'obre una nova porta de negoci.

#### 4- Marta Prim Sabrià

**Descripció:** Professora de la UAB (Universitat Autònoma de Barcelona).

**Responsabilitats:** S'encarregarà de coordinar el projecte de l'estudiant amb la finalitat de que aquest estudiant pugui avançar en les diverses fases del projecte, obtenint el major criteri d'èxit possible.

**Criteri d'èxit:** Aconsegueix coordinar perfectament en tot moment el projecte de l'estudiant i aquest n'obté excel·lents resultats.

#### 5- UAB (Universitat Autònoma de Barcelona)

**Descripció:** Universitat on l'estudiant realitza els seus estudis.

**Responsabilitats:** Proporciona els estudis a l'estudiant i permet que aquest pugui realitzar aquest projecte. Puntuarà el projecte de l'estudiant el dia que aquest estigui finalitzat, validat per la tutora (coordinadora) del projecte i es presenti davant d'un jurat.

**Criteri d'èxit:** L'alumne aprova el projecte.

#### 6- Jonathan Fragoso Pérez

**Descripció:** Estudiant d'enginyeria tècnica en informàtica de sistemes i desenvolupador Junior d'aplicacions en la plataforma Android.

**Responsabilitats:** S'encarregarà de dur a terme aquest projecte així com tota la part d'anàlisi, memòria, arquitectura i implementació un cop s'hagin definit i pactat els requeriments per part del client i el cap de projectes. Anirà coordinat en tot moment pel cap de projectes, amb la finalitat de que els resultats siguin el més bons possibles.

**Criteri d'èxit:** La solució aportada esdevé exitosa, la planificació del projecte es segueix i el pfc s'entrega a la data indicada finalitzant correctament tots els requeriments i funcionalitat, així com la documentació. L'empresa està contenta amb aquest desenvolupador i la seva resolució del projecte.

#### **USUARIS**

A continuació, l'altra part interessada són els usuaris que podran fer ús d'aquesta solució.

S'especifiquen els mateixos camps descrits anteriorment per a cadascun dels stakeholders.

## 1- Ajuntament de Barcelona

**Descripció:** Aquest grup utilitzarà l'aplicació de gestió que només ells tindran, la qual a través del seu ús, influirà directament en l'aplicació d'un usuari comú (ciutadà de Barcelona).

**Responsabilitats:** S'encarregaran de crear noves rutes en la ciutat, assignar els tags NFC corresponents a cada ruta, així com de gestionar la part de gamificació (premis, fites, etc). Sense aquesta gestió l'aplicació dels ciutadans no té sentit, ja que no hi haurà rutes per a practicar l'esport ni objectius.

**Criteri d'èxit:** La gestió és simple i ràpida, sense haver de fer un aprenentatge d'ús de l'aplicació.

## 2- Ciutadà de Barcelona

**Descripció:** Ciutadà que es descarrega l'aplicació per tal de fer ús dels serveis que ofereix. Tot i no ser el client que paga, podríem dir que és el client final, ja que utilitzarà el servei en forma d'aplicació pel qual paga el client.

**Responsabilitats:** Utilitzarà l'aplicació, la votarà i ajudarà a millorar-la reproduint bugs o millores en l'ús d'aquesta. Quan es registri, proporcionarà també les dades que vol l'Ajuntament.

**Criteri d'èxit:** L'aplicació li permet portar una millor gestió dels seus avenços físics practicant l'esport. L'aplicació li és senzilla d'utilitzar gràcies a una pantalla d'ajuda sobre la utilització de l'NFC (activitat de help en la interfície). A més la seva motivació creix gràcies al concepte de gamificació i xarxa social, poder interactuar amb amics, guanyar els seus temps, obtenir premis i fites, etc. Estarà content de l'ajuntament de la ciutat que ha llançat aquesta aplicació que per a ells és útil.

### 2.1.5. Entorn de treball de l'usuari

Com que hi ha dos aplicacions, una per administració de les etiquetes NFC i l'altra pels ciutadans, els entorns de treball d'aquestes seran diferents. No obstant, ambdues tenen un punt en comú i es tracta de que són aplicacions per mòbils i en aquest cas desenvolupades amb la plataforma Android.

És clar també, que tot i ser Android, el NFC, s'està començant a integrar des de fa un temps, i hi ha molts dispositius que no podran utilitzar aquesta aplicació, ja que en el seu hardware no contenen un xip NFC.

A més a més, cal tenir en compte també que les etiquetes NFC que s'enganxaran per a

fer els diferents controls de la ruta, han de ser d'un tipus específic, és a dir s'enganxaran en l'exterior, per tant només ens seran útils els tags amb aquesta característica, que podran superar les adversitats meteorològiques sense cap problema. Cal també pensar algun sistema per a protegir aquestes etiquetes i que la gent no les pugui desenganxar, ja que si així fos, el sistema deixaria de funcionar.

Donat que tenim dos tipus d'aplicacions diferents, els entorns seran els següents:

- \* Entorn de treball "aplicació administració": S'utilitzarà en el departament on hi hagi el grup que s'encarregarà d'afegir les noves rutes, assignant cada tag NFC a la ruta corresponent. Necessitaran dispositius amb Android i NFC, i tags buits en contingut per tal de que l'aplicació faci la seva feina (escriure la informació necessària en el tag i la base de dades per assignar aquest tag a la ruta escollida). Llavors aquestes etiquetes s'enganxaran al carrer al lloc al qual els pertanyi (l'escollit en la localització del tag).

- \* Entorn de treball "aplicació ciutadans": Els usuaris que tinguin Android i NFC, la podran descarregar al Google Play (mercat d'aplicacions de Google), i podran veure la informació que conté: rutes, usuaris, fites, premis, etc, des de qualsevol lloc, però quan facin ús de la "part esportiva" faran els "check in" a les etiquetes que l'ajuntament ja haurà enganxat.

### **2.1.6. Producte i documentació**

- L'aplicació dels ciutadans, es trobarà al mercat d'Android, per tant no és necessària una guia d'instal·lació, ja que únicament fent una cerca per nom es podrà descarregar des d'allà a través d'una connexió a Internet (ja sigui per wifi o internet mòbil).

- L'aplicació de gestió es subministrarà a l'ajuntament i no es publica al mercat d'Android, per tant una solució seria guardar l'aplicació (extensió .apk) en un dels nostres servidors, i posar en un document la url (enllaç) de descàrrega i un codi de barres (per llegir l'enllaç i no haver-lo d'escriure al dispositiu). Llavors es començarà a descarregar si el dispositiu està connectat a internet (ja sigui per wifi o internet mòbil) i un cop descarregada, suggerirà instal·lar aquesta aplicació al dispositiu.

### **2.1.7. Normativa i legislació**

Les normatives i lleis que cal tenir en compte per a dur a terme aquest projecte són les següents:

- Normativa de projectes de la UAB.

[http://www.uab.cat/Document/541/595/Normativa\\_PFCNovembre2010.pdf](http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf)

- Llei de protecció intel·lectual. <http://www.mcu.es/propiedadInt/index.html>

- LSSICE (Llei serveis de la societat de la informació i comerç electrònic).

<http://www.lssi.es/>

- LOPD (Llei orgànica de protecció de dades de caràcter personal).

<https://www.agpd.es/portalwebAGPD/index-ides-idphp.php>

- Llei d'enjudiciament criminal. <http://www.fmyv.es/ci/es/Ss/11.pdf>

## 2.2. Requeriments i restriccions

A continuació, s'avaluaran els requeriments propis d'aquest projecte. Aquests, es poden classificar en funcionals o no funcionals depenent de si són requeriments de la funcionalitat de la solució, o bé si influeixin sobre la manera com ha de funcionar aquest funcionalitat. S'esmenten cadascun dels requeriments de manera detallada i la prioritat d'aquests depenent de la importància que tenen. Degut a que la solució consta de dues aplicacions, es tractaran aquests requeriments separatament.

### 2.2.1. Requeriments funcionals

#### **APLICACIÓ DE GESTIÓ**

Inicialment cal dir que aquesta aplicació estarà formada de dues parts en el producte final i d'una per aquest projecte de final de carrera:

- D'una part en l'aplicació final tots els punts excepte el 2.2.1.3 es faran des d'una Web App (aplicació web). Aquest últim es gestionarà a través d'una aplicació amb Android, ja que per tal d'incloure la informació en l'etiqueta NFC, necessitem el xip NFC del dispositiu mòbil.

-D'altra banda, pel pfc, la primera part (Web App) es suprimirà i s'introduiran directament les dades en la base de dades.

**Prioritat:** Totes les característiques que s'esmenten ara en aquest apartat 2.2.1 (aplicació de gestió) són de prioritat alta, ja que són la base de l'aplicació, les que permeten introduir dades, per tant sense aquestes no té sentit el producte pels ciutadans.

**NOTA:** Les característiques que veurem a continuació, han de seguir un ordre cronològic; és a dir, la característica 2, no es pot implementar sense la 1 i així successivament.

#### 2.2.1.1. Creació i edició de rutes

Aquesta característica permet crear noves rutes en la ciutat on es durà a terme la pràctica de l'esport i es podrà utilitzar l'aplicació pels ciutadans. Aquesta característica consisteix en associar un nom a aquesta ruta, una descripció, així com els punts que formen la ruta (punts de control).

#### **2.2.1.2. Creació i edició de punts de control en les rutes**

Un cop s'han creat les rutes, cal associar els punts de control on l'usuari "marcarà" que ha passat per allà i s'actualitzaran els temps emprats en el recorregut. Aquests punts tenen associats unes coordenades que ens permetran veure al mapa aquest punt, una fotografia (opcional) per a veure el lloc on està posicionat el tag NFC (on s'haurà de fer el "check in"), una descripció i paràmetres opcionals com si en aquell lloc hi ha fonts per a beure aigua, monuments històrics, d'entre altres.

#### **2.2.1.3. Associar tags NFC als punts i escriure-hi la informació**

Ja s'han creat els punts de control dins de les rutes, ara el següent pas és associar aquests punts als tags NFC. Amb això volem dir posar informació dins de l'etiqueta: número de la ruta i el número de punt dins d'aquesta ruta, així com guardar l'identificador d'aquest tag a la base de dades, per tal de tenir-los controlats.

#### **2.2.1.4. Creació/edició de fites, events amb premis**

I finalment, per tal de gamificar l'experiència d'usuari, es podran crear fites, events i premis amb la finalitat de proporcionar auto-motivació i esperit de superació a l'esportista. Definirem fita, event i premi amb les següents característiques:

- Fita: Com el seu nom indica, es tracta d'un objectiu a realitzar. Seria un exemple de fita realitzar 4 rutes diferents de dificultat mitjana.
- Premi: El concepte de premi es basa en una fita premiada. Amb altres paraules, l'ajuntament parlarà amb col·laboradors, per tal de premiar a l'usuari si aconsegueix un objectiu concret. Un exemple d'això seria guanyar una entrada al cinema al que aconsegueixi obtenir el millor temps dins d'una data determinada i sobre una ruta.

### **APLICACIÓ DELS CIUTADANS**

**NOTA:** Els dos primers punts també segueixen un ordre cronològic.

#### **2.2.1.5. Veure les rutes disponibles**

Un cop s'inicia l'aplicació en el dispositiu, i després d'obtenir les dades mitjançant un "splash", es dirigeix a l'usuari a una pantalla (activitat en Android) on pot veure una llista de les rutes existents. Un cop es selecciona una ruta, es podrà veure el detall d'aquesta ruta amb la següent informació:

- Nom de ruta i tipus d'esport que s'hi practica (córrer, bicicleta o ambdós).
- Mapa amb els punts de control.
- Informació general.

- Dificultat i distància total.
- Llista de punts de control que inclou i ordre a seguir per a practicar-la.
- Millors temps: millor temps de ruta, millor temps del propi usuari de tots els cops que l'hagi completat (en cas de que no ho hagi fet mai, no es mostrarà aquesta dada).
- Afegir la ruta als favorits de l'usuari, m'agrada de Facebook, suggerir la ruta a un amic.

**Prioritat:** Alta

#### 2.2.1.6. Veure els punts de control associats a una ruta

Per tal de poder veure la informació d'un punt en concret, l'usuari farà un clic en l'element que vol veure de la llista de punts de control i l'aplicació es dirigirà al detall d'aquest punt. La informació d'un punt és la següent:

- Nom del punt i posició que ocupa (ordre) dins de la ruta.
- Imatge real (fotografia) del lloc on està posicionada l'etiqueta per a fer el "check-in".
- Informació del punt.
- Mapa amb la geolocalització del punt on a més hi haurà possible informació addicional:
  - \* Icones en cas de que sigui un punt on hi pot haver fonts, monuments històrics, lavabos públics, etc.
  - \* Un punt podrà ser de 3 tipus possible, depenent de l'ordre dins de la ruta:
    - CAS 1 (**Punt inici**): Punt on comença la ruta i per tant la pràctica de l'esport. Es mostrarà el punt posterior, per a saber cap a on s'ha de dirigir l'usuari.
    - CAS 2 (**Punt intermig**): Punt intermig en la ruta, podem veure des de quin punt ve l'usuari i el punt posterior (a on es dirigirà l'usuari després de passar per aquest control).
    - CAS 3 (**Punt final**): Punt on finalitza la pràctica de l'esport, es mostrarà també el punt anterior.

-M'agrada de Facebook.

**Prioritat:** Alta

#### 2.2.1.7. Creació de reptes

- Repte: Parlarem d'events o repte quan els nostres amics que tenim en la xarxa social, ens han proposat superar els seus límits de temps en una ruta determinada.
- **Prioritat:** La creació de reptes té prioritat mitjana/alta ja que és una funcionalitat que afecta directament a la gamificació, i per tant a la idea de motivar a l'esportista.

#### 2.2.1.8. Veure històric de pràctiques

Des d'aquesta pantalla, l'usuari podrà triar un mes i el tipus d'esport que va practicar, d'aquesta manera podrà veure tots els dies on va dur a terme aquest esport amb la informació corresponent i així veure el progrés.

**Prioritat:** Alta

#### 2.2.1.9. Gamificació d'events

-Abans de res donar una petita definició prèvia de gamificació:

*"La gamificación quiere hacer cualquier actividad más atractiva y emocionante, mediante el desarrollo de comportamientos deseados. Esta técnica puede animar a la gente para realizar tareas que consideran aburridas como completar encuestas, ir de compras o la lectura de páginas web."*

**Sergio Jiménez Arenas.** "Gamification"

Web: <http://www.gamkt.com/gamification-gamificacion/>

De manera transparent a l'usuari, quan aquest finalitzi una ruta, es comprovarà si ha complert algun dels objectius d'aquests events, i si és així s'informarà a l'usuari a més a més de reconduir-lo a la pantalla d'objectius i mostrar aquest objectiu com a marcat.

**Prioritat:** Mitjana/Alta

#### 2.2.1.10. Amics

Un cop un usuari es registra en l'aplicació i mitjançant una connexió a la API de Facebook, s'afegiran tots els amics associats al compte de l'usuari i que tenen instal·lada l'aplicació. Això permet fer ús de la gamificació d'events.

**Prioritat:** Mitjana

#### 2.2.1.11. Rutes preferides

Aquesta característica consisteix en guardar les rutes preferides de l'usuari i així aquestes seran accessibles a l'usuari de manera ràpida en algun lloc dins de l'aplicació.

**Prioritat:** Mitjana

#### 2.2.1.12. Check-in en un tag NFC

Fer un "check-in" en una etiqueta NFC implica apropar el dispositiu encarant la seva part de darrera (és on està localitzat el xip NFC) a l'etiqueta NFC fins que el dispositiu rebí la informació del tag i s'activi l'aplicació mostrant les dades corresponents al punt de control.

Les dades que es mostraran seran variables depenent del punt de control (veure diferents



tipus de punt de control a l'apartat 2.2.1.6), les opcions possibles són:

- Punt de control de tipus inici: Informa del nom de la ruta i permet mostrar el mapa per a veure els punts de control de la ruta. Després l'usuari escollirà l'esport que vol practicar abans d'iniciar la seva marxa.
- Punt de control intermig: Avisa del punt de control que s'ha assolit (ordre dins de la ruta) i convida a l'usuari a continuar fins a trobar el punt final.
- Punt de control de tipus final: Notificació de final de ruta, informa del conjunt de temps parcials obtinguts, temps total, distància recorreguda, calories, velocitats i altres dades informatives. Addicionalment, s'incorporen els objectius assolits (en cas de que se n'hagi superat algun) i s'ofereixen serveis:
  - \* Suggestir la ruta.
  - \* Repte: suggerir a un company superar el temps obtingut.
  - \* Afegir ruta a favorits.

**Prioritat:** Alta

#### **2.2.1.13. Informació durant el "tracking" de la pràctica**

Es mostra la localització actual de l'usuari i els punts de control per on ja ha passat, velocitat actual, distància recorreguda, temps parcial des de l'últim punt de control i temps final. S'ofereix la possibilitat de finalitzar la ruta, cas en el qual s'adverteix a l'usuari de que totes les dades de "tracking" d'aquesta pràctica es perdran.

**Prioritat:** Alta

#### **2.2.1.14. Preferències d'usuari**

Permet registrar un usuari així com activar o desactivar les notificacions que s'implementaran en un futur. També es podria associar un color al LED del dispositiu per quan es rebien aquestes notificacions.

**Prioritat:** Alta

#### **2.2.1.15. Ajuda de l'aplicació ("help" d'usuari)**

Petita pantalla que té com a finalitat facilitar a l'usuari l'aprenentatge d'ús de l'aplicació. Com que l'aplicació serà prou senzilla i intuïtiva, aquest help contindrà només informació d'ús del "check-in".

**Prioritat:** Alta

## **2.2.2. Requeriments no funcionals**

### **2.2.2.1. Navegabilitat i interfície d'usuari**

La navegabilitat i interfície d'usuari són requeriments no funcionals en ambdues aplicacions, ja que inclouen el com es mostraran aquestes dades a l'usuari per a aconseguir la millor eficiència i simplicitat.

**Prioritat:** Alta

### **2.2.2.2. Pantalla d'ajuda**

L'ajuda a l'usuari es mostrarà mitjançant un petit vídeo que simularà com s'utilitza l'aplicació, la funcionalitat bàsica, perquè aquest pugui començar des del principi a utilitzar-la sense cap problema.

**Prioritat:** Alta

### **2.2.2.3. Missatges descriptius d'error**

Informen a l'usuari dels problemes que hi pugui haver durant un "check-in" o, caigues del servidor, etc. Aquests missatges han de ser prou descriptius, senzills i gens tècnics, per tal de que siguin entenedors per a qualsevol persona. L'aplicació inclou 3 idiomes (català, castellà i anglès).

Aquests missatges seguiran la guia d'estil d'Android:

**Android developers style guide.** "Writing style"

Web: <http://developer.android.com/design/style/writing.html>

**Prioritat:** Alta

### **2.2.2.4. Disseny adaptat a les necessitats del client**

Malgrat com ja s'ha comentat anteriorment, Android té una guia d'estil per al desenvolupament de les aplicacions, el nostre client (Ajuntament de Barcelona) té la seva pròpia guia d'estil amb la finalitat de que totes les seves aplicacions segueixin unes pautes. Aquesta guia d'estil la podem trobar a:

**Guia d'estil gràfic de l'ajuntament per a aplicacions mòbils Android.** "Writing style"

Web: [http://www.bcn.cat/guiaestils/appmobils/android/android\\_estandard/welcome.html](http://www.bcn.cat/guiaestils/appmobils/android/android_estandard/welcome.html)

**Prioritat:** Alta

#### **2.2.2.5. Mapes de Google en 3D**

S'emprarà la última versió de mapes de Google en Android, ja que inclou mapes en 3D de les ciutats més grans del món. D'aquesta manera s'aconseguirà obtenir una major atracció a l'usuari.

**Prioritat:** Mitjana

#### **2.2.3. Restriccions del sistema**

Com a restriccions del sistema tenim:

- Restriccions en la interfície de l'aplicació seguint la guia d'estil de l'ajuntament.
- Re-avaluació dels requeriments del client per al pfc, és a dir reducció de la quantitat de la funcionalitat de l'aplicació respecte l'entrega final, per tal de poder assolir el plaç d'entrega i presentació d'aquest. Com ja s'ha dit anteriorment, les notificacions no estaran disponibles per a aquesta versió i la Web App tampoc es farà.

### **2.3. Alternatives al mercat i justificació de la solució**

Hi ha una gran quantitat d'aplicacions que ajuden a dur un control a l'usuari a l'hora de fer esport. Una de les més importants és Endomondo sports tracker.

Estem parlant d'una gran aplicació que es pot utilitzar amb una gran quantitat d'esports: córrer, bicicleta, bicicleta de muntanya, patinatge, diverses modalitats d'esquí, golf, surf, equitació i molts altres esports. És molt coneguda pels esportistes, ja que a més a més està disponible no només per a Android, sinó que també per a WP7 (Windows Phone 7), Blackberry, iOS. Es pot descarregar en dues modalitats: una gratuïta amb funcionalitat bàsica, i també en versió de pagament on augmenta el nombre de funcionalitats. A més a més es venen també uns receptors que connectats al dispositiu, permeten donar dades addicionals com la freqüència dels batecs del cor, amb gràfiques incloses. També es poden comprar opcions per separat com versió de baix consum, obtenir gràfiques des del mòbil, posar-se fites a un mateix per a superar-les, d'entre altres funcionalitats.

En definitiva una aplicació molt extensa i on els marges d'error són molt petits en termes de velocitat, distància (dependrà de la cobertura gps del dispositiu) per als que volen tenir-ho tot controlat a l'hora de fer esport.

Tap&Go no serà tan extensa, ja que es construirà a partir de les necessitats (requeriments) del client. Es vol un tracking en un lloc tancat, a la ciutat de Barcelona i on potser en un futur els usuaris podran suggerir noves rutes a l'ajuntament, però qui té la última paraula i qui crea aquestes rutes i associa les etiquetes és l'ajuntament.

La nostra aplicació intenta innovar, basant-se en la tecnologia NFC per a fer un tracking físic, tocant els tags i obtenint una major fiabilitat pels trams on el gps pugui funcionar de manera errònia. El concepte de gamificació integrat, també intenta fer que aquesta solució sigui diferent a la aportada pels competidors en el mercat. Una bona interfície pot ser el punt culminant de l'èxit.

## 2.4. Planificació

A continuació, s'introdueix la planificació que s'utilitzarà per tal de desenvolupar el projecte.

S'exposen els recursos, així com la seva disponibilitat, els possibles riscos que poden sorgir durant el desenvolupament i la reacció davant d'aquests, per tal de solucionar aquests problemes el més aviat possible, aconseguint que el temps final del projecte no es vegi afectat, o si ho fa, que l'efecte sigui el menor possible.

### 2.4.1. Descripció de la metodologia emprada: Scrum

La metodologia emprada per a dur un control del projecte és l'Scrum, una metodologia àgil de gestió de projectes que cada dia s'utilitza per a projectes de desenvolupament de SW.

El fet de pertànyer al grup de metodologies àgils, implica parlar doncs, d'una metodologia incremental on es divideix el projecte en petites parts funcionals i en donar més importància a l'equip de desenvolupament en general (que és qui treballarà en el projecte) i no tant al cap de projectes.

La feina del cap de projectes passa a ser la de donar totes les eines necessàries a l'equip perquè aquest projecte doni els millors resultats possibles. S'encarrega de que l'entorn de desenvolupament sigui òptim i hi hagi bona relació entre l'equip, així com de pensar com solucionar tots els problemes que pugui tenir l'equip durant la implementació i desenvolupament del producte.

Scrum, abraça el canvi; és a dir, en qualsevol moment en què el client vulgui nous requeriments o canvis, s'han de poder fer i de manera fàcil. Cal una bona gestió, ús d'eines com UML (Unified Modeling Language) per a obtenir una coordinació en l'equip, així com una bon disseny de l'arquitectura SW.

Aquesta metodologia inclou una nomenclatura depenent de la responsabilitat de les persones que treballen en el projecte i també per a les diferents fases que es volen distingir durant el desenvolupament.

#### **SPRINT**

És una fase, un increment en el desenvolupament del producte.

L'equip en l'inici del projecte defineix la durada dels sprints, normalment es recomana que

el període sigui de 2 setmanes, tot i que la durada podrà variar depenent de les condicions, del tipus de projecte i de l'experiència de l'equip.

Després, s'associa el contingut de cadascun d'ells; s'imposa l'objectiu: les activitats i tasques que hi haurà dins de cada sprint. Llavors es divideixen les tasques entre l'equip.

Cadascun dels sprints que tingui el projecte, suposarà que els objectius del projecte fins a aquell sprint funcionen, i per tant el client podrà veure una demostració totalment funcional del producte.

Cal ser realista i assegurar bé les tasques que entren dins de cada sprint, tenint en compte l'experiència de l'equip.

## **ROLS DE L'EQUIP**

Es poden distingir diferents papers d'acord a la responsabilitat i funció que tenen en el projecte:

- Product owner: S'encarrega de validar que fa l'equip de desenvolupament, s'adequa a les necessitats del client. És llavors qui s'encarrega del control comercial d'acord a la solució que s'ha demanat.

En aquest projecte, el cap de projectes i el comercial seran qui tindran aquest paper.

- Scrum Master: És el cap de projectes. No és el líder de l'equip, però proporciona totes les eines que l'equip de desenvolupament necessiti i de resoldre els conflictes que l'equip pugui tenir durant la implementació. A més, també s'encarrega de que les fases i el reglament de l' SCRUM es compleixin.

El cap de projectes ocuparà aquest paper.

- Equip de desenvolupament: Té la responsabilitat d'autogestionar-se per a aconseguir lliurar els objectius que es demanen per a cada sprint. No es recomana un equip massa gran, ja que és ideal un equip que es coneix, que es pot ajudar. L'equip ha de tenir la capacitat de respondre ràpidament als problemes i parlar d'aquests problemes ràpidament amb l'Scrum Master, per tal de seguir les dates dels sprints.

En aquest projecte l'equip està format per l'estudiant que realitza el projecte, però es relacionarà amb els dissenyadors (per a valorar i fer els canvis oportuns en el disseny de l'aplicació) i l'equip de sistemes (per a validar l'elaboració la base de dades i l'aplicació del servidor).

- Stakeholders: Es tracta dels clients, proveïdors, comercials i tota la gent que participa en el projecte però no durant la implementació, sinó que ho fan en les revisions de cada sprint.

Aquest paper l'ocuparà l'Ajuntament com a client.

- Directors: Estableix l'equip, l'entorn de desenvolupament i els rols de l'equip.

En aquest cas, la feina serà del cap de projectes.

## **MEETINGS O REUNIONS EN L'SCRUM**

Un cop més, el fet de ser una metodologia àgil, implica que la coordinació sigui òptima i per tant és essencial incloure moltes reunions entre l'equip. És per això que es fixen diferents tipus de reunions, d'acord al contingut, algunes d'elles que s'utilitzaran són les següents:

- Daily Scrum o daily: És una reunió que es fa diàriament amb l'equip de desenvolupament i l'Scrum Master. Cal fixar una hora concreta perquè sempre la reunió es faci a la mateixa hora (si es pot) i que l'equip sigui puntual. La durada d'aquesta reunió ha de ser d'entre 10 i 15 minuts.

S'exposa com va avançant dia a dia el projecte i els problemes que van sorgint durant el desenvolupament, per a actuar de manera ràpida.

- Reunió de planificació d'sprint: Es defineixen les tasques per al proper sprint, s'assignen a l'equip d'acord a l'experiència i es valoren les diferents possibilitats per a desenvolupar la solució, decidint entre tots la solució final.

- Reunió de revisió d'sprint: Quan acaba un sprint es valoren els resultats obtinguts, per tal d'aprendre i no cometre els mateixos errors en els següents sprints.

## **DOCUMENTACIÓ**

També és necessari documentar tot el què es parla en les reunions, així com les tasques, problemes i altre informació necessària. Alguns d'aquests documents són:

- Product backlog: És un document d'alt nivell que defineix el product owner (client) d'acord als requeriments i funcionalitat del projecte. És allò que es construirà durant els sprints. És per això que l'equip pot opinar i aportar altres coses, però només és el client qui pot modificar aquest document.

- Sprint backlog: Sovint es fa amb una cartolina penjada a la paret amb el nom dels components de l'equip de desenvolupament i amb una taula de tasques: per fer, pendents i finalitzades. S'assignen les tasques pertanyents a l'sprint a cada membre, s'escriuen en post-it i es posen a l'estat: per fer. Quan els membres comencen una tasca o la finalitzen mouen els post-it per tal d'obtenir la millor organització possible en cada moment.

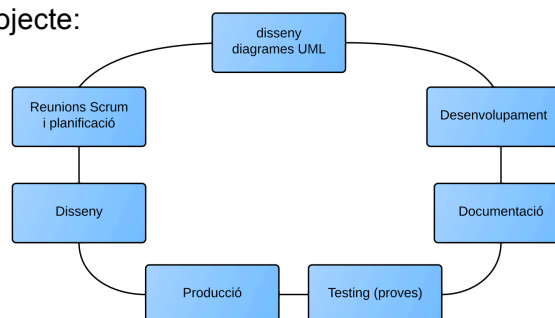
## 2.4.2. WBS (Work Breakdown Structure)

L'WBS (work breakdown structure) consisteix en enumerar les fases, activitats i fites del projecte, així com descriure-les, abans de fer el calendari i la planificació.

En l'Scrum es defineixen les tasques en cadascun dels Sprints. D'altra banda, sí que es pot definir inicialment unes fases dins del projecte:

- Fase d'inici i estudi de viabilitat: En aquesta fase el client defineix els objectius del producte. Un cop descrita la solució, s'inicia l'estudi de viabilitat amb la finalitat de veure si el projecte és viable o no per a l' empresa. Si ho és, es fa un primer esbós del disseny de la solució perquè aquest client el validi i demani canvis si ho desitja. Un cop està tot ben definit començarà la segona fase: desenvolupament del projecte.

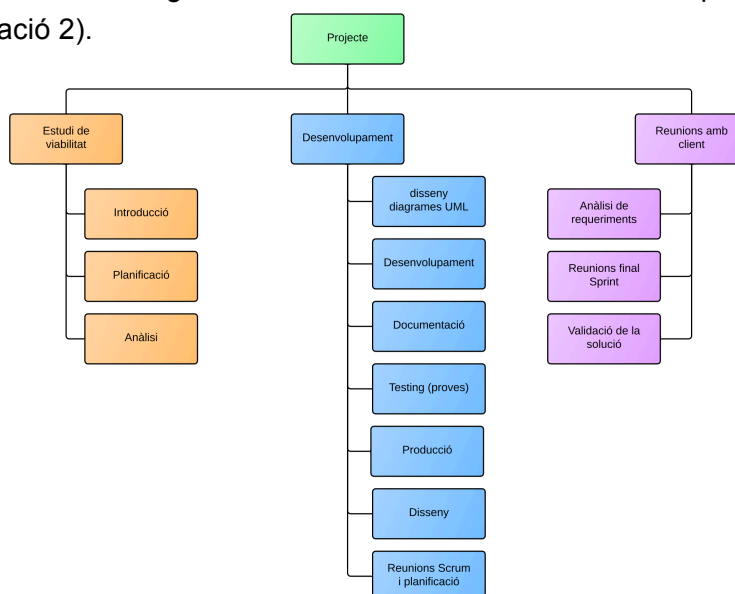
- Fase de desenvolupament: És aquesta fase on l'Scrum serà essencial i permetent abraçar nous canvis en el projecte (base de l'Scrum), es faran les activitats descrites a continuació en el següent diagrama (il·lustració 1). De forma incremental, i seguint l'ordre que marqui l'sprint corresponent a la fase del projecte:



Il·lustració 1: Estructura d'un sprint

Les fites (milestones) venen marcades pel contingut de cadascun dels sprints. Així doncs cada iteració o sprint és una fita dins del projecte, ja que és una part funcional del producte final.

A continuació es defineix detalladament el diagrama total de les activitats necessàries per a realitzar aquest projecte (il·lustració 2).



Il·lustració 2 : Diagrama WBS

### 2.4.3. Calendari

A causa de l'Scrum, no es pot fer un diagrama temporal posant les tasques que es realitzaran en cada moment. No obstant, sí que es pot fer un càlcul estimat del temps i exposar dependències de les activitats.

Això és perquè com es treballa mitjançant la iteració i per sprints, és impossible associar totes les tasques des de l'inici del projecte, la durada es marcarà en els sprints i aquests inclouen tasques.

Malgrat el què s'ha esmentat, es pot donar una estimació de temps ja que el projecte s'entrega al març i per tant és important coordinar bé el temps per obtenir aquesta data final. Estimarem els següents temps:

- **Fase d'anàlisi, estudi de viabilitat, disseny i UML:** Es va iniciar el 27 de Setembre, dedicant 6h diàries entre setmana i una mitjana de 4h diàries els caps de setmana. Aquesta fase finalitza el 30 de novembre, donant com a resultat un total de **289 hores** en total.

- La **fase de desenvolupament** comença el dia 1 de desembre i s'empraran 6h diàries i els caps de setmana 5h (si és necessari s'ampliarà la franja horària dels caps de setmana) fins al 14 de març. Obtenim un resultat de **588 hores**.

- Llavors obtenim una durada total de **877 hores** estimades per part de **l'estudiant**.

- Per part de la **tutora** que coordina el projecte de l'estudiant, el temps emprat són **9 hores**.

- Els **dissenyadors** validaran els wireframes de l'aplicació que farà l'estudiant. A més a més, també s'encarregaran de dissenyar les icones que s'empraran i d'escollir els colors i tipografia corresponents en cadascuna de les pantalles. S'estimen unes **40 hores** en total.

- El **tècnic de sistemes**, farà una introducció ràpida a l'estudiant de conceptes genèrics de servidors i de base de dades: back-end , així com de la generació de serveis. S'estima una jornada laboral de **8 hores**.

- El temps invertit pel product owner, no es podrà fixar ja que dependrà de la disponibilitat del client i és més difícil seguir la planificació d'Scrum.

- Finalment, manca estimar les hores del **project manager** (Scrum Master). Fins a la fase d'implementació, el temps emprat serà de 5h. Aquest temps inclou el brainstorming (pluja d'idees), definició de requeriments, contacte amb el client, validació de wireframes i planificació amb l'equip.

En la fase d'implementació, es farà el daily de 10 minuts, i la reunió de final de sprint de 30 minuts cada dues setmanes (durada de l'sprint). Obtenim un total de 11h del daily scrum i 3h 30 min de les reunions de final de sprint (on es definiran també els



requeriments del següent sprint).

Sumarem 2h i 30 min de problemes que puguin sorgir i de planificació de la presentació. Obtenim com a total **22 hores**.

## **2.4.4. Riscos del projecte**

### **2.4.4.1. Llistat de riscos i catalogació**

Durant el desenvolupament del projecte, poden sorgir alguns problemes, als quals si no es reacciona a temps, poden fer que la finalització del projecte s'endarrereixi o que sigui convenient retallar la llista de requeriments.

Malgrat alguns dels riscos que poden aparèixer no es poden controlar i s'hauran de valorar en el moment, alguns d'ells sí que els podem predir. Cal també pensar què es pot fer per resoldre'ls o assignar una probabilitat de succés i en l'impacte que poden tenir en el projecte.

A continuació (taula 1), s'exposen els riscos, descrivint-los, assignant una probabilitat de succés i impacte.

**Taula 1: Catalogació de riscos**

ID	Risc	Efecte	Impacte	Probabilitat
R1	Els sprints no es defineixen correctament.	Es van endarrerint les tasques a fer per al proper sprint. La durada final del projecte es pot veure afectada.	Crític / Catastròfic	Mitjana
R2	Canvis de requeriments	Tornar a planificar les tasques, i per tant els sprints. Les conseqüències que esdevenen a partir d'això són les de R1.	Crític / Catastròfic	Mitjana
R3	Incompliment de normativa	El producte no podrà donar a la llum, ja que hi poden haver multes a causa d'aquest incompliment.	Catastròfic	Baixa
R4	No es realitzen suficients backups (còpies de seguretat)	Es perdran totes les parts de les quals no s'hagi fet còpia de seguretat. Per tant, s'haurà de tornar a fer tot i la data fi	Crític / Catastròfic	Baixa

		variarà.		
R5	Testing incorrecte	La consistència del producte no serà bona i possiblement la gent es desinstal·larà l'aplicació i no esperarà a una nova actualització (pèrdua de clients).	Catastròfic	Baixa
R6	Eines de desenvolupament inadequades	El temps s'allargarà o bé la qualitat del producte no serà tant bona.	Crític	Baixa

#### 2.4.4.2. Pla de contingència

Cal elaborar un pla de contingència per tal de resoldre aquests problemes en cas de que sorgeixin i així, tenir-los controlats, perquè les conseqüències siguin mínimes.

En la següent taula (taula 2) s'exposa una solució a cadascun d'aquests riscos.

**Taula 2: Pla de contingència**

ID	Solució
R1	Tornar a planificar l'sprint o bé retallar funcionalitat, ja que el temps és la data d'entrega del projecte i no pot variar.
R2	Tornar a planificar l'sprint o bé afegir funcionalitat o canvis per al proper sprint.
R3	Replantejar el producte adaptant-lo per a complir la normativa.
R4	Utilitzar repositori de codi i fer còpies constants de tota la documentació i del projecte.
R5	Assegurar-se de que el testing sigui correcte i que el testegi l'equip de testing de l'empresa.
R6	Buscar primerament les eines oportunes encara que s'hagi de perdre temps informant-se, ja que després s'agrairà i no caldrà fer les coses dos cops.

## 2.5. Càlcul de costos (pressupost)

Cal fer un anàlisi consistent dels costos econòmics per tal de veure si aquest projecte és viable econòmicament per a l'empresa i per tant val la pena ésser realitzat. És per això que es tindran en compte les diferents estimacions que es detallen a continuació.

### 2.5.1. Costos directes o de personal

En aquest apartat es descriuen els costos unitaris del personal implicat en aquest projecte i després el cost total d'acord al temps emprat.

En la taula següent es descriuen els costos aproximats dels recursos de personal del projecte, el temps emprat en cadascun dels recursos i el total parcial en relació a aquests. Cal tenir en compte que són costos aproximats i obtinguts de dades de mitjanes, ja que els costos reals no els podem saber (ja que el projecte es desenvolupa en l'empresa).

En la següent taula (taula 3) es descriuen els costos directes, quantificant el cost unitari del personal i el parcial a conseqüència del temps i d'aquest cost unitari.

*Taula 3: Recursos directes*

Recursos	Cost / h	Temps	Total (euros)
Desenvolupador	10 euros	877h	8770
Scrum Master	30 euros	22h	660
Product owner	25 euros	---	---
Dissenyador	24 euros	40h	960
Tècnic sistemes	24 euros	8h	192

Obtenim un total de **10.582 euros** de costos directes.

### 2.5.2. Costos indirectes: material i llicències

I ara, en la següent taula (taula 4 a la següent pàgina) es detallen els costos unitaris del material implicat. També es detalla el període d'amortització i el període d'ús de cadascun dels components, afegint el cost d'amortització.

**Taula 4: Recursos indirectes**

Recursos	Cost unitari	Període d'amortització	Període d'ús	Cost d'amortització
PC Dell (oficina)	1000 euros	3 anys	6 mesos	167 euros
Macbook (2007)	---	---	---	Ordinador de l'estudiant
MS Windows 7 Professional	309 euros	4 anys	6 mesos	38,62 euros
MS Office 2010	Llicència d'estudiant	---	---	---
NFC starter pack kit	46,06 euros	---	---	46,06 euros
MagicDraw Professional Java license	--- (han proporcionat llicència gratuïta fins al febrer a l'estudiant)	4 mesos	4 mesos	---
Lucid chart license	9,95 euros	2 mesos	2 mesos	19,9 euros

Els costos totals **indirectes** obtinguts a través de la taula anterior són de **271,58 euros**. Cal tenir en compte també altres costos propis de l'empresa que no sabem com el cost del repositori SVN, material d'impressió del projecte i altres.

### **2.5.3. Resum i anàlisi cost-benefici**

El **total de costos directes** ( 10852 euros) i dels **indirectes** (271,58 euros) és de **11123,58 euros**.

Un cop analitzats el total de costos necessaris per al desenvolupament del projecte, l'empresa ha fet una valoració i resol que aquest projecte és viable, ja que la relació cost/benefici és bona i per tant la viabilitat econòmica també esdevé validada.

Els nombres finals del benefici no es poden donar, en tractar-se d'un projecte d'una empresa.

## **2.6. Conclusions de viabilitat**

Un cop realitzat l'estudi, s'arriba a la conclusió que el projecte és totalment viable en tots els àmbits: els objectius es poden assolir, la posició al mercat del producte pot ser molt bona, i l'elaboració de la planificació i els beneficis econòmics també fan viables el projecte.



## 3. Disseny de l'arquitectura SW: UML

### 3.1. Introducció i eines de disseny

Un cop s'ha fet un estudi i es tenen clars els requeriments del projecte, s'ha de pensar en com solucionar-lo. En el disseny de SW, l'UML (Unified Modeling Language) és essencial per a treballar en equip i per a descriure l'arquitectura de l'aplicació, ja que és un llenguatge universal que conté la seva pròpia normativa, patrons i especificació dels diagrames; per tant és entenedor per a tothom. Això implica que, si s'han de fer canvis en el producte, qualsevol persona que hi hagi de treballar, es pugui fer una idea general de com funciona allò, sense haver de llegir i entendre milers de línies de codi.

### 3.2. Document d'arquitectura

Per a aquest projecte, s'han elaborat un conjunt de diagrames UML. Alguns d'aquests documents tenen les il·lustracions annexades a la memòria, per tal de millorar la visibilitat ja que tenen unes dimensions considerables i la reducció implicaria no observar-los correctament. En els diagrames podrem observar que hi ha integrada la WebApp que s'implementarà en un futur, per tal de poder afegir rutes, events i usuaris a partir de taules, així com consultar les dades sense que sigui necessari accedir a les taules de la base de dades per a fer aquestes operacions.

A continuació es detallen els diagrames emprats:

#### 3.2.1. Diagrama de casos d'ús

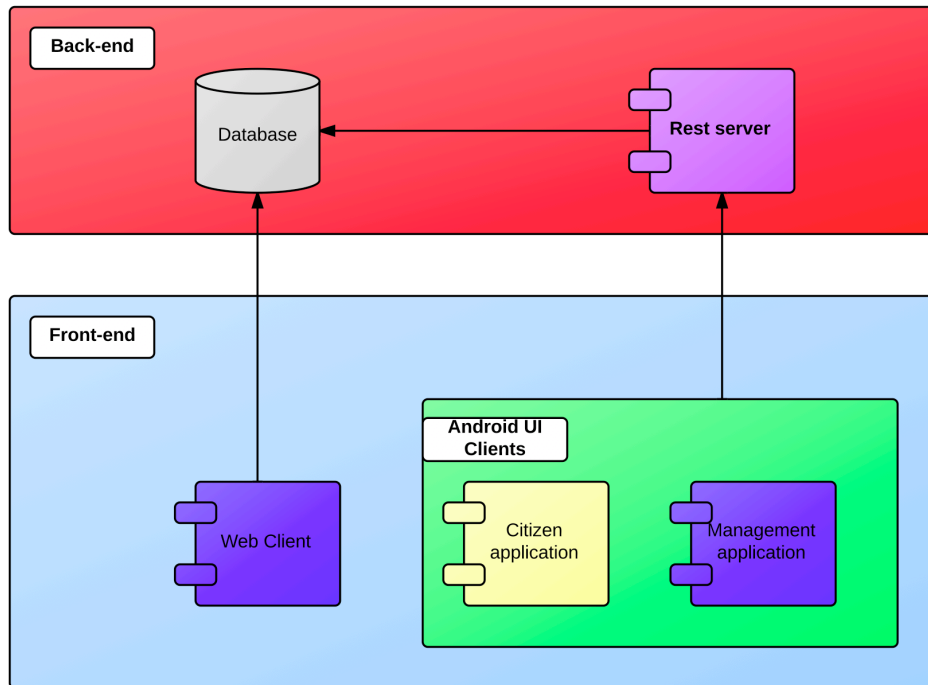
Especifica a nivell d'usuari, com aquest usarà cadascun dels requeriments de l'aplicació. Com tenim dues aplicacions, hi ha dos documents on cadascun d'ells especifica la funcionalitat pròpia d'un usuari d'aquell tipus d'aplicació. Els diagrames es troben en l'annex (diagrames apartats 1.1 i 1.2 respectivament) i s'hi veuen reflexats els requeriments exposats en l'estudi de viabilitat anteriorment, però més orientats cap a l'ús dels usuaris dins de l'aplicació.

#### 3.2.2. Diagrama de components

El producte, es basa en l'arquitectura client-servidor. S'han emprat 2 diagrames de components per a especificar aquest tipus d'arquitectura:

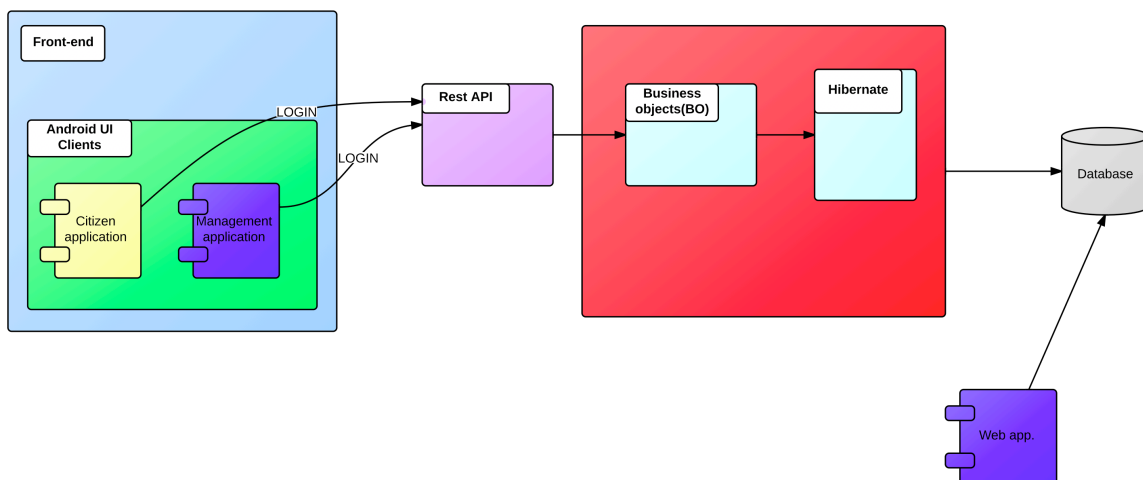
- \* Diagrama de components genèric: Especifica de manera genèrica com el front-end (part client) accedeix al contingut de la base de dades mitjançant el servidor (back-end). La part client consta de 2 parts: l'aplicació de gestió per assignar els tags NFC i les fotos i l'aplicació dels ciutadans.

Caldrà preparar els serveis de la API del servidor, aplicant seguretat i controlant que cada aplicació accedeixi als seus propis serveis i no pugui accedir als de l'altre. En la següent il·lustració (il·lustració 3) podem veure el funcionament:



**Il·lustració 3: Diagrama de components genèric**

\* Diagrama de components del servidor: Conté l'arquitectura de components del servidor. La construcció del servidor es farà mitjançant Servlets amb Java i això implica l'ús dels BO (Business objects) que permeten i faciliten l'accés i emmagatzemat de les dades a la base de dades. L'API limitarà els serveis als quals podrà accedir cadascun dels clients. En la següent il·lustració (il·lustració 4) podem veure aquest funcionament.



**Il·lustració 4: Diagrama de components del servidor**



### **3.2.3. Diagrama de paquets**

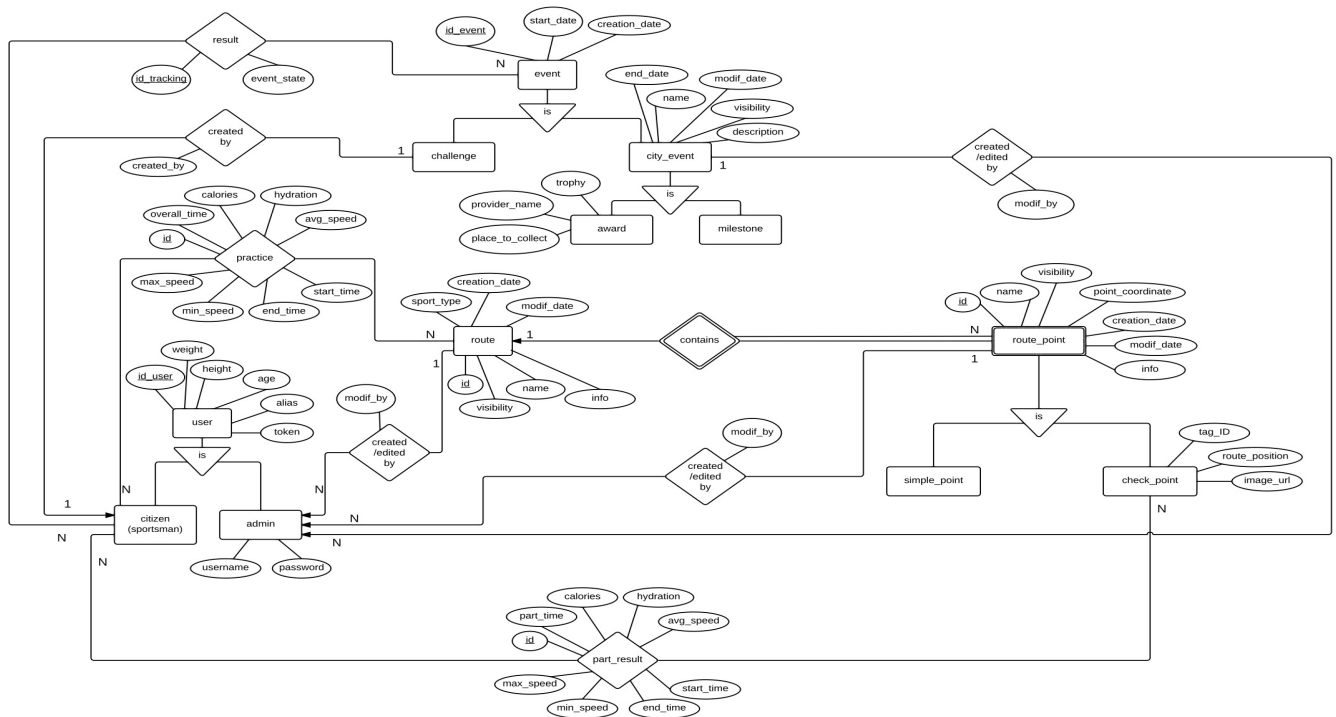
- **Diagrames de paquets**: Permeten especificar a nivell genèric en quins packages (paquets) s'agruparan les classes a l'hora de desenvolupar l'aplicació. Permeten una millor organització i separació de components depenent de la funcionalitat. Un cop més tenim dos diagrames en l'annex (1.3 i 1.4). Es basen en l'arquitectura de MVC (model/view/controller) o arquitectura de model/vista/controlador. Cadascun dels paquets especifica el seu contingut en termes de funcionalitat.

### **3.2.4. Diagrama de classes**

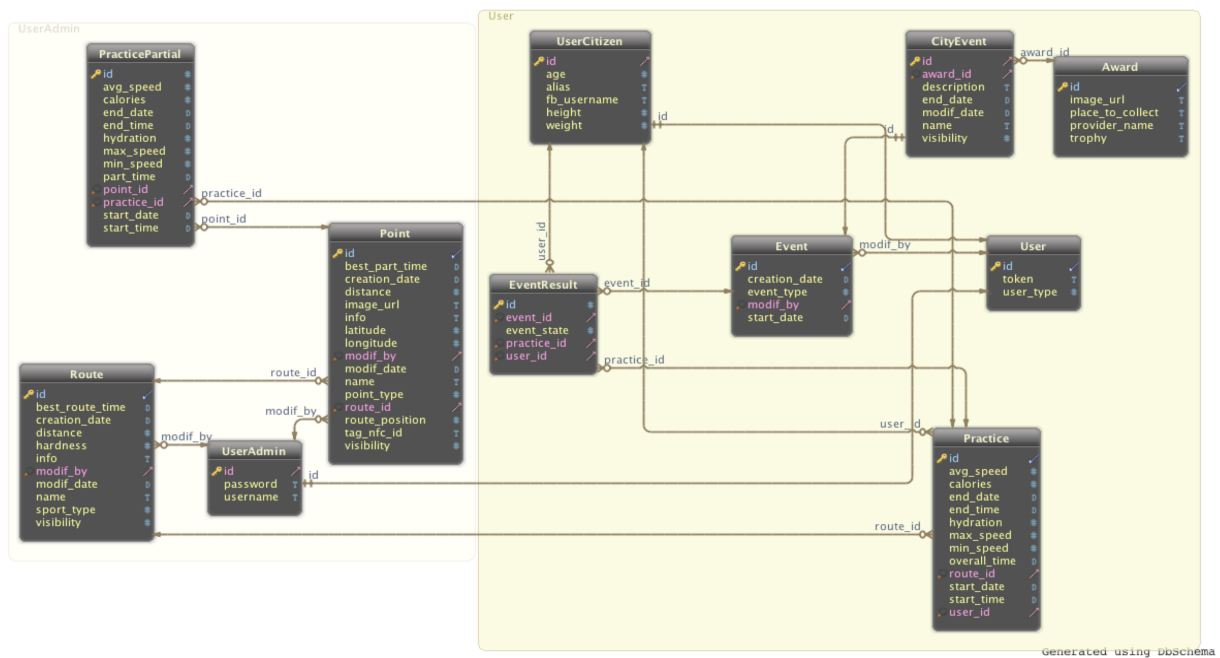
- **Diagrama de classes**: Inicialment s'elabora un diagrama genèric de classes i mitjançant l'eina MagicDraw UML es generen les classes per a començar a desenvolupar i programar. S'aniran afegint elements quan s'escaigui durant els sprints (recordar que l'Scrum és iteratiu i per tant permet canvis en qualsevol moment) i al final es farà enginyeria inversa utilitzant de nou el programa descrit anteriorment, per a obtenir un diagrama de classes genèric i final. Aquest diagrama està contingut dins de l'annex (1.5-Diagrama de classes).

## **3.3. Disseny de la base de dades**

Per finalitzar tot el disseny de diagrames amb UML, es crea el disseny de la base de dades. A continuació podrem veure el diagrama entitat/relació i les taules finals de la base de dades. No es creu necessari explicar el funcionament, ja que les taules i les seves relacions defineixen per sí mateixes el funcionament. En continuació podrem veure les il·lustracions 5 i 6 de com funciona l'arquitectura de la nostra base de dades. Es mostren en la següent pàgina per tal de amb més claredat i detall les imatges.



II-lustració 5: Disseny entitat/relació de la base de dades



II-lustració 6: Taules de la base de dades

## 4. Disseny de la UI

Un cop s'ha realitzat l'estudi de l'arquitectura i dels components, es pot passar a la creació del disseny de la interfície. En el cas de les aplicacions mòbils, una bona interfície és essencial. En molts casos, aquesta tindrà un pes de 60% (a vegades superior) de l'èxit de l'aplicació.

El disseny d'aplicacions mòbils és molt complexe, ja que no només depèn de l'organització d'imatges, colors, tipografies, etc, també depèn de la navegació en les pantalles, de la simplicitat d'ús per a nous usuaris, però alhora incorporació d'elements d'accés ràpid pels usuaris més avançats, d'usar els propis components que proporciona el sistema operatiu, però també innovar per a captar l'atenció de l'usuari. És important que els textos i els missatges descriptius d'error siguin curts i entenedors, amb la intenció de no limitar el nombre d'usuaris que l'utilitzin. Un factor més de dificultat és la col·locació dels elements, ja que l'ordre destaca prioritats i, per tant, què es vol donar a transmetre; la idea general de cadascuna de les pantalles.

Quan es tracta de solucions a demanda, aquest disseny s'adapta totalment al client, afegint encara més complexitat, ocasionada per l'estil propi que vol el client. En aquest cas, l'ajuntament de Barcelona té la seva guia d'estil propi pel disseny d'aplicacions per a dispositius mòbils. Aleshores, addicionalment de seguir la navegabilitat i l'estil d'Android, s'incorporaran els colors, tipografia, icones i altres elements propis d'aquesta guia del client.

A continuació, es mostren els primers esbossos que es van fer del disseny, i després el disseny final d'ambdues aplicacions, especificant a alt nivell l'idea general de cadascuna de les pantalles. Després en l'annex podrem veure el resultat final i veure la comparació.

### 4.1. Sketches (primers esbossos del disseny)

Inicialment, l'estudiant va començar a fer els esbossos a mà, però després de fer diversos canvis a petició del departament de disseny, es va creure més convenient aprendre una base d'Illustrator i Photoshop (Adobe CS4), per tal d'anar realitzant els canvis, sense haver de re-dibuixar a mà tota la pantalla per cadascun d'ells.

El resultat final d'aquests esbossos, està contingut dins l'annex (apartats 2.1 i 2.2).

## 4.2. Wireframes (disseny final)

Després dels esbossos, s'inicia la fase final de disseny. Els elements (colors, botons, imatges, tipografia, diàlegs, etc) que s'incorporen en aquesta, formaran part de l'aplicació final.

Un cop els dissenyadors han realitzat aquest disseny final, es valora, es mostra a l'equip i es comenta en què consisteix cadascuna de les parts, perquè quedi totalment clar com funciona aquesta interfície i no afecti al desenvolupament a conseqüència de no haver-la entès bé. Es proporciona un document amb les tipografies, codis dels colors (html), les dimensions dels marges entre components i les imatges finals.

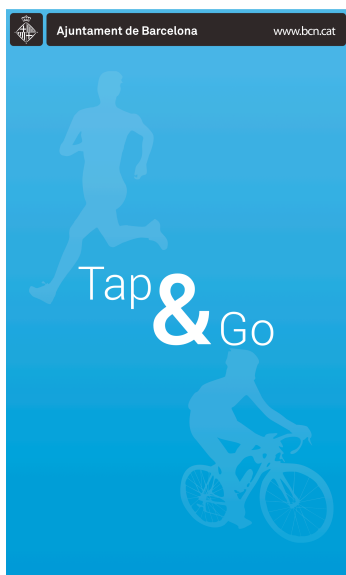
Ara, es descriuran les diferents pantalles (activitats o activitats en Android) mostrant-les i afegint una breu descripció (en l'annex podem veure el resultat final; apartats 2.3 i 2.4), ja que després en la implementació es detallarà més el seu funcionament:

Il·lustració 7: icona de l'aplicació

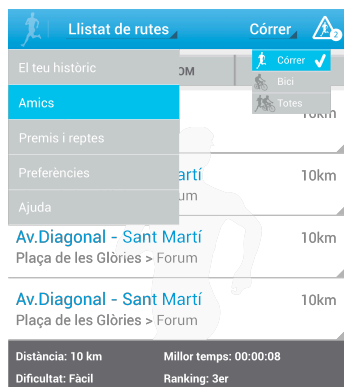


### APLICACIÓ CIUTADANS

Il·lustració 8: Splash activity



Il·lustració 9: Rutes



Il·lustració 10: Detall de la ruta



**Splash:** Pantalla inicial de càrrega que mostra l'eslògan de l'aplicació.

**Rutes:** Es carrega després de l'splash. Mostra informació de les routes disponibles, permet filtrar-les i hi trobem els menús que podrem utilitzar per moure'ns dins de l'aplicació.

**Detall de la ruta:** Un cop s'ha seleccionat una ruta, es mostra l'informació d'aquesta, així com el mapa, els punts de control que la formen i elements de gamificació.

## Il·lustració 11: Detall del punt



**Detall del punt:** Després de triar un punt d'una ruta, podem consultar els paràmetres d'aquest punt, un mapa, una imatge de la localització de l'etiqueta i una panoràmica.

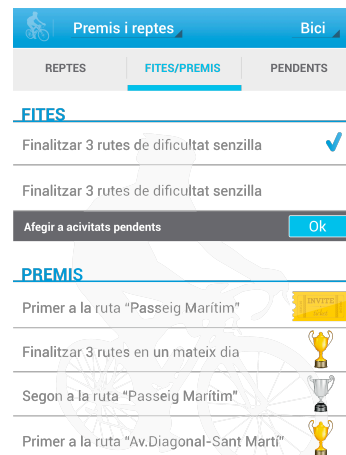
**Històric:** Aquí, l'usuari podrà consultar les pràctiques que ha anat realitzant des de que va instal·lar l'aplicació i filtrar-les per diferents paràmetres.

**Premis i reptes:** Activitat on es centralitzen tots els events que té un usuari i el seu estat.

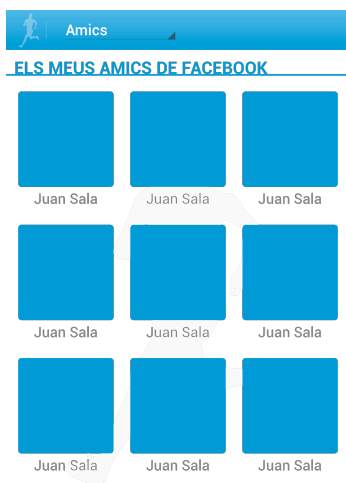
## Il·lustració 12: Històric



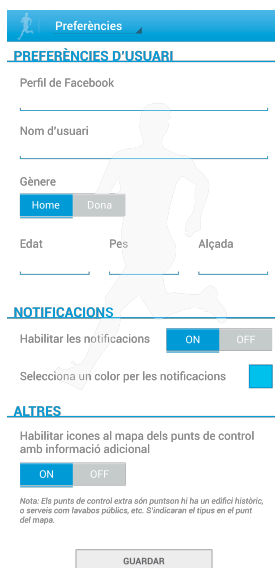
## Il·lustració 13: Premis i reptes



## Il·lustració 14: Amics



## Il·lustració 15: Preferències



## Il·lustració 16: Inici de pràctica



**Amics:** Mostra un llistat dels amics de l'usuari a Facebook i que utilitzen Tap&Go

**Preferències:** Menú de preferències de l'usuari. Conté el formulari de registre i opcions que funcionaran en un futur (notificacions i icones addicionals als mapes).

**Inici de la pràctica:** Inici de pràctica en una ruta. Es selecciona l'esport.

## II-lustració 17: Pràctica

Passeig Marítim

1 Platja de la Mar Bella

Platja de la Mar Bella

Punt de control validat satisfactòriament!

ACCEPTAR

Velocitat	Temps parcial
2,5 km/h	00:10:16
Distància recorreguda	Temps total
1200 m	00:11:30

FINALITZAR

**Pràctica:** Aquesta activitat mostra tots els avenços de l'usuari al llarg de la pràctica.

**Resum de pràctica:** Mostra els resultats parcials obtinguts i altres dades i convida a fer ús de la gamificació

## II-lustració 18: Resum de pràctica

Passeig Marítim

RESUM DE LA PRÀCTICA

Felicitats!

Ha assolit l'últim punt de la ruta. A continuació es mostraran els seus resultats en aquesta pràctica:

1 Platja de la Mar Bella	00:01:30
2 Platja de la Mar Bella	00:02:30
3 Bogatell	00:02:30
Nova Icària	00:02:35
TOTAL	00:09:05

Velocitat mitja	Velocitat màxima	Calories
2,5 km/h	10,5 km/h	312 kcal

COMPARTEIX RESULTATS RETAR A UN AMIC

## APLICACIÓ ADMINISTRADOR

Conté la pantalla de rutes (amb un filtre per a mostrar les rutes invisibles als ciutadans), la pantalla de detall de ruta i finalment la del punt de control. Aquesta és diferent a la de l'aplicació dels ciutadans ja que permet assignar una etiqueta NFC al punt de control i afegir una foto del lloc, així com una panoràmica. També té un login d'administrador a l'inici.

## II-lustració 19:

Login admin.

Ajuntament de Barcelona www.bcn.cat

User Name

Password

ACCEPTAR

## II-lustració 20:

edició d'un punt

Passeig Marítim

Platja de la Mar Bella

INFORMACIÓ

Des d'aquesta ruta, l'esportista gaudirà de les vistes d'algunes de les platges de la ciutat.

ID del Tag NFC assignat: ASSIGNAR ETIQUETA

ASSIGNAR FOTO

GUARDAR

## 5. El servidor i l'API Rest

L'accés als serveis, s'ha desenvolupat mitjançant una petita API Rest basada en un conjunt de Servlets en Java.

Mitjançant la crida a aquests serveis des de les aplicacions mòbils, s'aconsegueix accedir a les dades de la base de dades i, s'obté el control i separació de dades entre l'aplicació client i la dels ciutadans.

És important saber que aquest API Rest és provisional i específic pel projecte de l'estudiant, ja que el pes del projecte recau en les aplicacions mòbils. Un cop es vulgui implantar la solució final, el servidor es tornarà a dissenyar i desenvolupar, i posteriorment, s'instal·larà en les instal·lacions de l'ajuntament. És per això, que actualment el servidor està instal·lat en un Tomcat en local a la màquina de l'estudiant i el dispositiu mòbil es connecta a la IP assignada al ordinador en la xarxa.

A continuació, s'exposa la relació que hi ha entre les diferents pantalles en l'aplicació i les peticions al servidor (crides als Servlets). Després, es detallen a nivell genèric els Servlets, els mètodes que contenen i la seva funcionalitat.

### 5.1. Relació events de la UI/peticions al servidor

A continuació, es detalla la relació que hi ha entre les diferents pantalles en l'aplicació i les peticions al servidor. Es separa, l'aplicació d'administració de l'aplicació dels ciutadans.

#### **APLICACIÓ CIUTADANS**

Les peticions que es fan en l'aplicació són les següents:

- Obtenir les rutes, pràctiques i events: En la pantalla inicial del llistat de rutes, es descarregaran aquestes dades per tal de fer aquesta descàrrega un únic cop i navegar sense "loaders" (element de càrrega en la interfície) posteriorment. Les peticions es faran als següents servlets segons l'ordre abans esmentat: RoutesServlet, PracticesServlet, EventsServlet.
- Informació d'una única ruta: Usant l'NFC es pot arribar a accedir al detall d'una ruta directament. Això implica no passar per la pantalla inicial del llistat on es carreguen les rutes. Per tant cal obtenir la informació a partir del id de ruta que s'obté de l'etiqueta NFC. La petició es farà al RouteServlet.
- Descarregar imatges obtingudes del servidor: Els punts de control tenen una imatge que correspon a la localització de l'etiqueta NFC i també una imatge panoràmica que permet veure amb més detall l'entorn. Si una imatge no està encara carregada al dispositiu, es fa una petició per obtenir-la. La petició es fa al ImageDownloadServlet.

- Formulari de registre d'un usuari: Des de les preferències, un usuari es podrà registrar per tal d'utilitzar la gamificació i participar en els events. El registre es fa mitjançant el FormServlet.
- Desar a la base de dades una pràctica: Quan l'usuari finalitza una pràctica i si està registrat, té la possibilitat de desar-la al servidor, amb la finalitat de participar en els events. Si en aquell moment no té connexió a Internet disponible, alternativament, podrà pujar la pràctica des de l'històric. La petició es farà al UploadPracticesServlet.
- Gamificació; reptar a un amic i "like" d'un lloc: Els ciutadans que estiguin registrats, podran enviar un repte a algun dels seus amics que també estiguin registrats dins de l'aplicació i que tinguin Facebook, amb la finalitat de que intentin batre els seus temps. També podran marcar que els hi agrada una ruta. Els servlets encarregats d'això són l'EventsUploadServlet i el LikesServlet.

### **APLICACIÓ ADMINISTRADOR**

Des d'aquesta es faran les següents peticions:

- Login d'administrador: Després de l'splash de càrrega de l'aplicació, apareix la pantalla de login on l'usuari administrador haurà de posar les seves credencials per tal de continuar. La petició de login es farà enviant els paràmetres que veurem després al LoginServlet.
- Rutes: En l'activitat del llistat de routes (apareix després de l'splash i el login), es demanen totes les routes disponibles a la base de dades. La petició es farà al RoutesServlet, passant addicionalment el paràmetre per obtenir les routes no visibles als ciutadans.
- Pujar imatges i identificador del tag NFC d'un punt: Malgrat són dos serveis diferents, ambdós es criden des de la pantalla d'edició d'un punt de control. Els Servlets que s'encarregaran d'actuar en funció d'aquestes peticions són el ImageUploadServlet i el PointServlet respectivament.

## **5.2. Implementació del servidor**

Aquest apartat, especifica els Servlets per accedir als serveis que necessiten les aplicacions. Addicionalment, s'afegeix la informació dels mètodes interns que criden aquests servlets per accedir a les dades de la base de dades i proporcionar la informació sol·licitada pels clients (aplicacions Android). Aquesta informació d'arquitectura entre el front-end (client) i el back-end (part servidor) es correspon amb el diagrama de components genèric que hem vist anteriorment (apartat 3.2.2).

Primerament, abans d'explicar el funcionament de tots aquests serveis, s'especifica la url o (enllaç) base per tal de que el client pugui connectar amb al servidor:



[http://IP\\_ASSIGNADA:8080/tapandgo/](http://IP_ASSIGNADA:8080/tapandgo/) on IP\_ASSIGNADA és la IP assignada a l'ordinador de l'estudiant dins de la xarxa on està connectat, 8080 correspon al port i tapandgo és el nom del directori on hi ha els arxius pertanyents a l'API Rest.

A partir d'aquesta url base es concatena el nom d'accés als servlets i s'envien els paràmetres escaients en cada cas. Totes les peticions retornen un objecte JSON de resposta, que variarà depenent del tipus de petició i que després, es parseja des de l'aplicació per a obtenir els resultats o controlar els errors (en cas que n'hi hagin).

A continuació es defineixen els servlets emprats:

### **TapAndGoServlet**

**Url del servlet:** --

**Paràmetres:** --

**Descripció:** Aquest servlet és especial. És el servlet base utilitzat per tots els servlets de Tap&Go, únicament s'encarrega d'obrir i tancar la base de dades cada cop que es fa una petició i de controlar l'autenticació quan el servei requereix autenticació d'usuari. Tots els servlets hereten d'aquest, i així "s'oblidaran" de realitzar la connexió amb la base de dades i de controlar aquesta autenticació, i realitzaran només les operacions per a obtenir les dades i les retornaran en el format correcte (objecte json) al client.

### **EventsServlet**

**Url del servlet:** url base + **/events**

**Paràmetres:** event\_type (*int*), user\_id (*int*)

**Descripció:** Rep l'id de l'usuari que vol consultar els seus events, així com el tipus d'events que vol obtenir. La petició es fa per POST (són dades privades de l'usuari) i retorna un Json amb el conjunt d'events que ha sol·licitat l'usuari (reptes, fites o premis).

### **FormServlet**

**Url del servlet:** url base + **/form**

**Paràmetres:** fb\_username (*String*), alias (*String*), user\_id (*int*), height (*float*), weight (*float*), age (*int*), gender (*int*)

**Descripció:** Rep les dades d'un usuari per POST i les tracta: crea un usuari si no existeix, el modifica si les dades són diferents o bé l'esborra si l'usuari es vol donar de baixa com a usuari dins l'aplicació.

### **ImageDownloadServlet**

**Url del servlet:** url base + **/point\_images**

**Paràmetres:** image\_name (*String*), mime\_type (*String*)

**Descripció:** Rep el nom de la imatge a descarregar i el seu tipus (el mime type descriu el tipus de dades així com l'extensió). Cerca aquesta imatge en el directori on es desen les imatges i retorna un fitxer que conté la imatge. La petició es fa per GET.

### **ImageUploadServlet**

**Url del servlet:** url base + **/image**

**Paràmetres:** point\_id (*int*), file\_name (*String*), admin\_id (*id*), upload-file (*File*), image-type (*int*), mime-type (*int*)

**Descripció:** Permet que un usuari administrador, pugui pujar una imatge al servidor. En la base de dades es desarà la ruta de les dues imatges (imatge del detall, imatge panoràmica), així que és necessari saber quin tipus d'imatge es puja en cada cas (per això tenim el paràmetre image-type). La petició es farà per POST.

### **LikesServlet**

**Url del servlet:** url base + **/likes**

**Paràmetres:** place (*String*), request\_type (*int*), place\_id (*int*), user\_id (*int*)

**Descripció:** Aquest Servlet permet desar els llocs (rutes o punts de control) que li agraden a un usuari. Es simula un event de "like" a Facebook, per tal de saber quines són les rutes i punts que més agraden als ciutadans. D'aquesta manera, es sabrà quins són els millors llocs per a crear events i la tipologia de rutes que més agrada. La petició es farà per POST.

### **LoginServlet**

**Url del servlet:** url base + **/login**

**Paràmetres comuns:** user\_type (*int*), username (*String*)

**Paràmetres addicionals aplicació administració:** password (*String*)

**Paràmetres addicionals aplicació client:** user\_id (*int*)

**Descripció:** S'encarrega del login d'un usuari genèric o administrador. Malgrat el login s'inicia en el mateix Servlet, aquest es fa en mètodes separats en el DAO (data access object) ja que depenent del paràmetre user\_type, el servlet envia la petició al mètode que s'escau. Es fa evident utilitzar una petició POST per qüestions de seguretat de l'usuari.

### **PointServlet**

**Url del servlet:** url base + **/point**

**Paràmetres:** point\_id (*int*), nfc\_tag\_id (*String*)

**Descripció:** Permet a un usuari administrador pujar a la base de dades l'identificador únic

del tag NFC (cada etiqueta NFC té un id únic) associat al punt de control. La petició es fa per POST.

### **PracticesServlet**

**Url del servlet:** url base + **/practices**

**Paràmetres:** user\_id (*int*), username (*String*)

**Descripció:** Recupera totes les pràctiques que ha dut a terme un usuari. La petició es fa per POST.

### **RouteServlet**

**Url del servlet:** url base + **/route**

**Paràmetres:** id (*int*)

**Descripció:** Obté la informació de la ruta que conté l'identificador que es passa per paràmetre. La petició es fa per GET.

### **RoutesServlet**

**Url del servlet:** url base + **/routes**

**Paràmetres:** admin\_mode (*String*)

**Descripció:** Recupera el conjunt de rutes que hi ha desades a la base de dades i que són visibles per als ciutadans (ja estan creades i llestes per a fer esport en elles). Si, addicionalment, s'afegeix el paràmetre admin\_mode, llavors es recuperaran també les rutes no visibles; les que estan en procés de preparació per a que les puguin utilitzar els ciutadans. La petició es fa per GET.

### **UploadEventsServlet**

**Url del servlet:** url base + **/event**

**Paràmetres:** user\_id (*int*), to\_user (*int*), route\_id (*int*)

**Descripció:** Gràcies a aquest Servlet, podran enviar reptes als seus amics. La petició es fa per POST.

### **UploadPracticesServlet**

**Url del servlet:** url base + **/practice**

**Paràmetres:** user\_id (*int*), conjunts de paràmetres propis de la pràctica

**Descripció:** Permet pujar totes les dades parcials i totals d'una pràctica que ha fet un usuari. Si l'usuari supera algun repte, l'estat d'aquest també s'actualitza. La petició es fa per POST per a millorar la seguretat d'aquestes dades.



## 6. Desenvolupament de les aplicacions mòbils

Finalment, després d'un llarg procés d'estudi, de disseny mitjançant UML, un cop dissenyada la interfície i s'ha preparat la base de dades i el servidor per a proporcionar les dades a les aplicacions Android, comença la fase de desenvolupament i codificació.

Es desenvoluparà emprant Java com a llenguatge de programació i addicionalment, s'utilitza l'Android SDK i algunes llibreries que s'esmentaran en els sprints.

Es codificarà en Eclipse, un IDE molt conegut i segurament el més utilitzat per a tots els desenvolupadors que treballen amb Java.

Per tal de dur un control de versions, així com gestionar millor els canvis que s'aniran fent al llarg del projecte, s'utilitza SVN (Subversion) com a sistema de control de versions del codi. El repositori emprat és BeanStalkApp, que a més a més té la seva pròpia interfície web que reflexa els que s'han pujat en aquest repositori.

En aquest capítol s'exposaran les restriccions a nivell d'implementació i codificació, així com l'estructura a seguir per a codificar i finalment el desenvolupament del projecte, dividint les tasques en sprints.

### 6.1. Restriccions i estructura de codificació

Amb la finalitat d'obtenir una millor organització, preparada per abraçar el canvi (idea fonamental de la metodologia Scrum), es seguiran una sèrie de criteris en la nomenclatura dels layouts (xml on es codifiquen les vistes en Android), de les imatges, els identificadors de color i de les cadenes dels missatges que apareixen en l'aplicació.

Primerament, degut a que l'aplicació dels ciutadans estarà disponible en 3 llengües (català, castellà i anglès), és important consultar com s'envien les ordres a un usuari; és a dir, les formes verbals correctes en el camp del SW per cadascuna d'aquestes llengües. Els missatges que apareixen en l'aplicació, ja siguin d'error, botons, diàlegs, explicacions, etc, hauran de ser prou descriptius i entenedors perquè qualsevol persona els pugui entendre, per tant no s'usaran tecnicismes i si n'hi ha, s'especificarà la definició. A continuació es descriuran algunes d'aquestes regles a seguir:

- En Android, les cadenes de caràcters estàtiques es desen al directori *res/values-LANGUAGE/strings.xml* on LANGUAGE són les inicials de la llengua. En aquest cas tindríem *values-en*, *values-es* i *values-ca*. Aquesta separació, permet que sigui el propi sistema operatiu qui s'encarregui de mirar la llengua per defecte que té el dispositiu i, a partir d'aquesta, cercar les cadenes al directori corresponent en l'aplicació. En el cas que el dispositiu tingui una llengua diferent a les que abarca l'aplicació, tenim un directori per defecte que és *values* (aquest no conté guió i les inicials de la llengua) i les cadenes

s'agafaran per defecte d'aquest directori. Per tant, finalment tindrem els directoris *values*, *values-es*, *values-ca* mostrant per defecte les cadenes amb anglès en cas de que la llengua del dispositiu no sigui ni el català ni el castellà.

En la següent taula (taula 5), es descriu la nomenclatura pels identificadors de cadenes:

**Taula 5: Nomenclatura dels identificadors de cadenes estàtiques**

Nom	Descripció
error_TYPE	Els missatges d'error començaran amb error i aniran seguits d'una barra baixa i del tipus d'error.
dialog_TYPE_title	Pels títols dels diàlegs s'emprarà la nomenclatura dialog, el tipus del diàleg i title indicant que es tracta d'un títol.
dialog_TYPE_content	Es tractarà de la mateixa manera que l'anterior element, però finalitzant amb content, indicant que és tracta del contingut o descripció del diàleg.
loading_TYPE_text	En el cas de les cadenes que anuncien càrrega de dades, es descriuen amb loading i el tipus de dades que s'estan carregant.
preference_TYPE_title	Es tracta de la descripció o títol de la preferència i el tipus.
button_SCREEN_TYPE_text	El text que mostren els botons, s'identificaràn per button, el nom de la pantalla i el tipus de botó.

- Els colors es codificaran amb el corresponent codi HTML i es desaran de la següent forma: tapandgo\_COLOR\_DESCRIPTION on COLOR\_DESCRIPTION es correspon amb una descripció del color.

- A conseqüència de que el sistema operatiu suporta diferents tipus de pantalla, aquest suporta una estructura semblant a l'esmentada abans pels missatges. En aquest cas, les imatges es desen a *res/drawable-SCREEN\_RESOLUTION* on SCREEN\_RESOLUTION és la resolució suportada per la pantalla. Les proves es faran en un Samsung Galaxy Nexus (dispositiu de l'estudiant), amb resolució *xhdpi* (extra-high density), i les mides es treballaran amb *dp* (mida de treball per punts per densitat i per tant, independent als píxels) ja que treballant d'aquesta manera, s'aconsegueix adaptar les mides a diferents pantalles. Aleshores, les imatges es desaran al directori *res/drawable/xhdpi*.

Depenent de la tipologia de l'imatge, és a dir si és una icona, un fons, el fons d'un botó seleccionat o no seleccionat, un splash, etc, es segueix una nomenclatura que facilita la cerca durant el desenvolupament i que facilita la interacció entre els dissenyadors durant les modificacions que puguin sorgir durant el projecte. En la següent taula (taula 6) es descriu aquesta nomenclatura:

**Taula 6: Nomenclatura de les imatges**

Nom	Descripció
btn_selected_bkg_TYPE / btn_unselected_bkg_TYPE	Indiquen el fons que es fa servir per a un botó d'un tipus concret. Aquest fons serà diferent quan el botó està premut o no.
actionbar_icon_TYPE	Nomenclatura per als icones de l'action bar (barra d'accions superior d'Android).
icon_TYPE	Designa una icona que no pertany a l'action bar i la seva tipologia.
dialog_header_bkg/ dialog_content_bkg	S'utilitzen pel fons del títol/contingut en un diàleg.
filter_POSITION_selected_bkg / filter_POSITION_unselected_bkg	Fons seleccionat/no seleccionat pels diferents filtres superiors (depenen de la posició a la pantalla) utilitzats en el llistat de rutes i l'històric.
menu_item_selected / menu_item_unselected	Fons de color pels ítems dels menus superiors quan aquests estan seleccionats/no seleccionats.
sportsman_SPORT_TYPE_bkg	Es tracta del 3 fons diferents (bicicleta/córrer/ambdós) que mostren en el fons de la pantalla el filtrar per tipus d'esport.
etext_TYPE_bkg	Fons que utilitzen els elements de la interfície de tipus edit text (elements on es pot escriure).

- Els fitxers amb extensió .xml dels *layout*, també han de seguir una estructura en el seu nom. Aquest nom especifica per sí mateix si la vista correspon a una vista d'una pantalla concreta, si es tracta d'un ítem d'una llista, o bé el *layout* d'un diàleg, etc. En la següent taula (taula 6) es descriu aquesta nomenclatura:

*Taula 7: Nomenclatura dels layout*

Nom	Descripció
menu_item_TYPE	Descriuen el layout d'un ítem que pertany a un menú.
SCREEN_layout	Layout general d'una pantalla en concret.
SCREEN_item_list_header / SCREEN_item_list_footer	En les pantalles on tenim una llista i aquesta llista conté una capçalera i un peu de pàgina.
SCREEN_list_item	Defineixen la vista d'un ítem concret dins d'una llista en una pantalla.
dialog_TYPE	Emprat pels layouts dels diàlegs.
view_TYPE	Pels layouts de les vistes custom (vistes personalitzades) s'indica que és per una vista i la tipologia.

- El codi s'haurà de documentar amb Javadoc, estarà ben identat per tal de que sigui més entenedor, i finalment es seguiran les restriccions de Java en la nomenclatura de classes, mètodes, atributs, etc.

- D'altra banda, malgrat i que l'aplicació s'acabarà utilitzant a Barcelona, el vídeo i el test es farà en una ruta que es crearà a Parets del Vallès (lloc de residència de l'estudiant), ja que serà necessari anar diverses vegades a fer tests sobre la ruta per optimitzar velocitats, localització i temps.

- El model utilitzat durant tot el desenvolupament de les aplicacions, és el *MVC* (Model/view/controller) que consisteix a separar lògica, dades i interfície, per tal de que cada component realitzi només la seva funció. Les aplicacions necessiten obtenir dades (model), mostrar-les (view) i dur el control de l'interacció entre les accions que fa l'usuari i els events que succeeixen com a conseqüència en l'aplicació (controller).

## 6.2. El desenvolupament (Sprints)

Un cop aquesta normativa prèvia ja està definida, es pot començar el desenvolupament.

Ara es divideixen les tasques del projecte i s'agrupen en sprints. Aquest projecte s'ha dividit en set sprints amb una durada de dues setmanes cadascun. El final d'un sprint, implica tenir una part del projecte totalment funcional, és a dir, tindrem una part de l'aplicació que es pot mostrar al client, ja que aquella part es dona per finalitzada (sempre i



quan s'han complert tots els objectius).

A continuació s'especifica cadascun d'aquests sprints, exposant un breu resum de la funcionalitat que inclouen, la planificació de les tasques, es detalla la implementació, la fase de test i resolució de *bugs* (correcció d'errors) i es valora el resultat. La valoració inclou l'assoliment dels objectius, si la planificació inicial s'ha acabat seguint, els problemes i, en cas de que hagin afectat a la planificació o als objectius, com es solucionaran per tal de que el temps final del projecte no es vegi afectat.

### **6.2.1. Sprint 1**

En aquest sprint, es crearan les classes per tal de desar les dades que s'obtindran del servidor (aquestes classes s'anomenen *beans*) i després es crearà l'estructura general de la interfície en les pantalles de primer nivell. No es donarà funcionalitat als components, però sí que es prepararan per a mostrar les dades que volem mostrar. Per tant, això implica de manera indirecta, generar i modificar classes amb la finalitat de no haver de fer canvis en aquesta interfície un cop s'afegeixi la lògica en els seus components.

#### **a) Planificació**

- i) Generació dels beans corresponents. 6 hores
- ii) Estructura de la interfície en totes les pantalles de primer nivell, del menú i de les vistes personalitzades (custom views) necessàries. 66 hores
- iii) Importar en el projecte totes les imatges que s'aniran utilitzant, així com escriure totes les cadenes de caràcters en les 3 llengües. 2 hores
- iv) Preparació dels diferents tipus de diàlegs. 5 hores
- v) Creació dels filtres. 1 hores

#### **b) Implementació**

- i) A partir de les dades que es volen mostrar en les diferents pantalles i de la informació que s'obté del servidor, es creen els *beans*, que són classes contenidores de dades que tenen els seus atributs privats, als quals s'accedeix i es modifica el contingut a través dels *getters* i *setters*. A més alguns d'aquests objectes comparteixen atributs, degut a que són del mateix tipus, per tant i per tal de no repetir informació, farem ús de l'herència. Un exemple d'aquesta herència són els events que tots comparteixen unes característiques, però en tenen altres que els fan diferenciar. Es genera una classe genèrica que es diu Event i llavors CityEvent, ChallengeEvent i AwardEvent hereden d'aquesta classe afegint cadascun d'ells les seves característiques pròpies que els diferencien entre ells.
- ii) Abans de detallar la resolució de la tasca d'aquest sprint, s'exposen alguns conceptes previs relatius a la plataforma d'Android, ja que s'usaran diverses vegades al llarg de tota la fase d'implementació en els diferents sprints.

- Activity (activitat) --> Prèviament es dona la definició en anglès de la pàgina oficial de desenvolupadors d'Android:

*"An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View)"*

Web: <http://developer.android.com/reference/android/app/Activity.html>

D'aquesta informació podem extreure que al sistema operatiu d'Android, una activitat es tracta d'una pantalla, ja que conté la vista que es mostra per pantalla, així com els components que permeten interactuar amb l'usuari per tal de que aquest pugui realitzar accions i veure com aquestes modifiquen el comportament de l'aplicació.

- Layout --> És un fitxer en format xml, i que per tant agrupa la informació en diferents nivells, que conté les vistes estàtiques que carregarà una activitat, un diàleg o bé una vista personalitzada.

- Adapter --> La definició d'un adapter per part de la web de desenvolupadors d'Android és la següent:

*An Adapter object acts as a bridge between an AdapterView and the underlying data for that view. The Adapter provides access to the data items. The Adapter is also responsible for making a View for each item in the data set.*

Web: <http://developer.android.com/reference/android/widget/Adapter.html>

Un adapter doncs, és un component que rep un conjunt de ítems, i a partir d'aquests genera la vista de cadascun d'ells d'acord a la tipologia d'aquests i el comportament desitjat. Un adapter permet per exemple "inflar" una a una les vistes dels ítems d'un llistat i mostrar la informació desitjada en cadascun d'ells.

Aquestes són les definicions més importants a tenir en compte abans d'especificar la resolució dels punts pertanyents a la tasca d'aquest sprint. El resultat s'exposa a continuació:

- Menú superior: Ens permet moure'ns al llarg de l'aplicació en totes les pantalles de primer nivell que són totes aquelles que mostren informació prèvia, separada per temes i que proporcionen la idea essencial de l'aplicació. En aquest cas tenim el llistat de rutes, l'històric, la pantalla d'events, amics i ajuda. La pantalla de preferències s'incorpora en aquest menú, tot i ser de segon nivell, ja que proporciona el formulari d'inscripció d'un usuari en l'aplicació i aquest afecta a la funcionalitat i les dades que es mostren.

Aquest tipus de menú s'anomena *DropDownMenu* i el generem a través d'un adapter que aconseguim el llistat de cadenes dels ítems i successivament crea les vistes del menú. Aquesta vista del menú s'afegeix posteriorment a l'*action bar* (barra d'accions superior) i un cop afegida, s'informa a la barra d'accions de que la navegació és del tipus `NAVIGATION_MODE_LIST` a través del mètode `setNavigationMode(actionBarMode)`.

- Pantalles de primer nivell i custom views: Un cop s'ha creat el menú, que ens permetrà moure'ns per les diferents activitats de primer nivell, es passa a crear els xml amb les vistes de totes aquestes pantalles. Quan s'han generat, s'incorporen en el mètode `setContentView(layoutID)` de cadascuna de les activitats corresponents i s'aplica l'estil d'acord als wireframes del disseny. Llavors es crea una classe que anomenem `SingletonTest`, que segueix el patró *Singleton*, i en la qual afegirem llistats de rutes, events, amics, etc, per tal de mostrar les dades correctament per pantalla. Posteriorment, en l'sprint pertanyent, aquestes dades es recuperaran del servidor i de Facebook, però ens és necessari tenir un mitjà per a comprovar que les dades es mostren correctament.

Per a obtenir els llistats es generen els adapters corresponents a cada llistat, els quals reben les dades, les tracten i les fan mostrar en cadascun dels ítems del llistat.

Al menú superior, podem trobar també un filtre per tipus d'esport, que s'infla emprant una vista expandible, anomenada *Spinner* en android i seguint un procés similar al del menú, però sense afegir l'*actionMode* (ja que es tracta d'un filtre i no de la navegació de la pantalla), s'afegeix aquest spinner. Al marge dret de la barra també trobem una icona, que s'anomena *actionItem* en l'action bar, que mostra el nombre d'activitats pendents i reptes que té l'usuari i que es tracta d'un accés directe que un cop s'ha premut, ens porta directament a la pantalla d'events per tal de veure aquests events pendents que té l'usuari.

Totes les activitats que s'inicien, ja sigui des del menú, com des de les vistes que trobem a la pantalla, es criden amb el mètode `startActivity(context, class)` on *context* és el contexte actual de l'aplicació per afegir a la pila d'activitats del sistema operatiu la nova activitat i *class* és el nom de la classe de l'activitat que es vol iniciar.

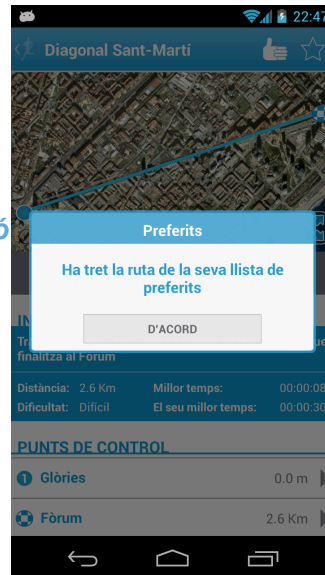
iii) Aquesta fase consisteix en anar afegint les imatges i missatges dins dels directoris corresponents. Això es fa seguint la nomenclatura que hem vist anteriorment; així com les regles esmentades per a la descripció dels missatges.

iv) En l'aplicació tenim 4 tipus de diàlegs diferents: els diàlegs informatius o afirmatius, diàlegs on l'usuari pot cancel·lar l'acció, el diàleg dels amics de Facebook i finalment el diàleg del calendari a la pantalla de l'històric. En aquest sprint prepararem els dos primers, mentre que els altres s'implementaran posteriorment, ja que pel diàleg dels amics ens cal haver integrat Facebook amb l'aplicació i el del calendari utilitza una vista de calendari pròpia integrant una llibreria de *CalendarView* i la complexitat en ser més elevada el nombre d'hores s'incrementa i no s'arribaria a temps a finalitzar l'sprint.

A continuació, es mostren els diàlegs que s'implementaran en aquest sprint (il·lustracions 21 i 22):

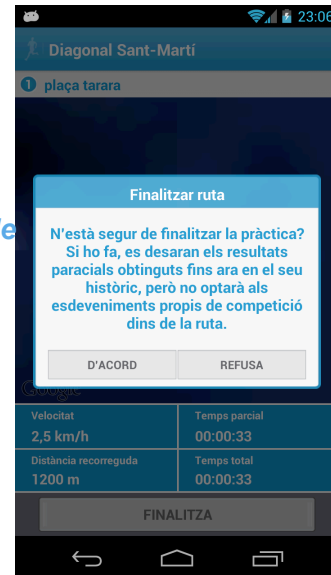
*Il·lustració 21:*

*Diàleg d'afirmació*



*Il·lustració 22:*

*Diàleg cancel·lable*



Ambós diàlegs contenen un títol, un contingut i botons. Degut a que aquests dos tipus de diàlegs s'aniran utilitzant al llarg de l'aplicació, es crea una classe per a cada diàleg, la qual rep el títol i el contingut per paràmetre. Un cop un objecte de tipus diàleg infla la vista amb aquests paràmetres, es prepara la funcionalitat dels botons: en el cas del diàleg d'afirmació, el botó afirmatiu tanca el diàleg i no té cap altra funcionalitat, en canvi, en el diàleg cancel·lable cal afegir dos listeners que avisaran de la decisió que ha pres l'usuari i actuar d'acord a aquesta.

✓ Els filtres permeten filtrar pels diversos paràmetres que podem trobar al llarg de les diferents pantalles. Els botons permetran interactuar amb l'usuari i un cop clicats, caldrà avisar a l'activitat per tal de que faci el filtratge i refresqui la pantalla per a mostrar els resultats.

La solució final consisteix en fer un altre *custom view* que conté la barra sencera que amb els 3 filtres. Aquesta classe té com a nom *SortSelector* i quan es crea, s'encarrega de crear la vista del selector de l'esquerra, intermig i de la dreta, ja que com tenen degradats i bordejats diferents, s'han de tractar de manera diferent. La fletxa descendent indica que els ítems de la llista estan ordenats en aquest ordre.

Les pantalles del llistat de rutes, l'històric i events, incorporen un objecte *sortSelector* (tal i com ja hem pogut veure als wireframes en l'apartat de disseny).

La classe integra el mètode *addSortButton(Filter filter, int viewType, Context context)* que permetrà afegir un filtre a cadascun d'aquests botons, d'aquesta manera, fem que la vista sigui indiferent al tipus de filtre que es vulgui afegir. Quan es crida a aquest mètode, la vista afegeix el text al botó, afegeix un tag únic dins l'aplicació a aquesta vista i s'actualitza.

L'usuari vol filtrar els ítems, és per això que necessitarem gestionar el clic en cadascun d'aquests filtres. És per això que la classe conté un *listener* (una interfície que avisarà a l'activitat) i aquest té el mètode `onSortSelectorPressed(String viewTag, int sortType)`, que es cridarà quan es clica al filtre i per tant, l'activitat que incorpori la barra de filtres s'assebentarà i cridarà aquest mètode també (aquest és el funcionament del patró *listener*). Llavors, l'activitat serà capaç de saber quin és el filtre que s'ha d'aplicar, a través del valor del `viewTag` que és l'identificador del filtre, i si s'ha de filtrar ascendentment o descendentment veient el valor de `sortType` (1 o 0 respectivament).

### **c) Test i valoració d'Sprint**

**Test:** Degut a que les dades encara no s'obtenen del servidor, es crea una classe que genera i desa objectes de manera estàtica per a veure que les dades es mostren correctament i que no falta informació de cap tipus en les pantalles. Es prova que els filtres funcionin i que la informació en pantalla es modifiqui un cop s'apliquen aquests filtres.

**Valoració:** Els resultats d'aquest sprint han estat els esperats, ja que el temps invertit ha estat l'esmentat en la planificació, i les tasques i objectius s'han provat satisfactòriament.

## **6.2.2. Sprint 2**

Un cop elaborades totes les pantalles de primer nivell (l'estructura o interfície d'aquestes), es passa a crear el detall de les rutes i dels punts control que formen part de cadascuna d'aquestes. La funcionalitat d'aquest sprint es centra en mostrar aquesta informació i els diferents tipus de mapes depenent del tipus i de si el mapa es veu a pantalla completa o no.

### **a) Planificació**

- i) Mapes. 38 hores
- ii) Pantalla de detall de ruta i de punt de control. 41 hores
- iii) Integració de l'sprint 1 i aquestes noves pantalles: navegació de la pantalla del llistat de rutes al detall d'aquesta i des d'aquesta última pantalla, l'accés al punt de control. 2 hores

### **b) Implementació**

- i) Els mapes, són una de les parts més importants d'aquest projecte, ja que són necessaris per a localitzar les rutes, els seus punts i durant el tracking. És per això, que es decideix aprendre com funciona la nova API v2 de mapes de Google en Android, que addicionalment a les novetats en l'arquitectura interna de la llibreria, incorpora mapes en 3D en les ciutats més importants del món, com és el nostre cas amb la ciutat de Barcelona.

Els mapes nous s'incorporen dins del *framework Google Service*, que té altres llibreries com les notificacions per push al núvol (*Google Cloud Messaging*), *Google Analytics*, *in-app billing*, etc. Per a utilitzar Google Services, és necessari primerament crear una API key, una clau única que informarà a Google que l'aplicació que estem desenvolupant utilitza els seus serveis.

Un cop s'ha obtingut l'API key, s'afegeix al manifest del projecte i ja es poden començar a utilitzar aquests serveis, en el nostre cas els mapes.

Llavors comença el procés de documentació del funcionament dels mapes, com s'afegeixen punts dins d'aquests i pel pintat de les rutes. També s'estudia com animar-los, i el posicionament de la càmera per a veure la ruta.

Finalment, es generen 4 mapes diferents dins de l'aplicació: un pel detall de la ruta, un pel detall d'un punt i els 2 mapes a pantalla completa. Els dos primers, no permeten que l'usuari pugui fer gestos, actuen com si es tractés d'una fotografia i no permeten fer zoom ni moure la càmera, ja que en ésser petits no es creu convenient per a l'usuari. En canvi, els altres dos mapes a pantalla completa sí que integren aquests gestos.

Per a pintar els punts es fa servir el mètode `addMarker` passant la posició on es vol pintar aquest punt i la imatge que es vol mostrar. Per a pintar la ruta, es fa servir el mètode `addPolyline` que va pintant una línia entre els dos punts que se li passen per paràmetre.

Aquests mapes s'utilitzaran a continuació en la següent part de l'sprint: el detall de la ruta i dels punts de control.

ii) Les pantalles del detall de la ruta i dels punts de control pertanyents a una ruta, inclouen un mapa de localització, informació relativa al lloc i altres paràmetres com el temps obtingut per l'usuari (si ha practicat aquesta ruta).

Prèviament a accedir a qualsevol d'aquestes dues pantalles ja s'ha obtingut la informació que es mostrarà. Això és perquè el llistat de rutes i el seu conjunt de punts es descarrega inicialment en la pantalla del llistat de rutes. Tota la informació de la ruta o bé del punt, així com el contingut del mapa s'omplen a partir d'aquesta informació.

A la part superior de la pantalla s'hi troba un botó de m'agrada, mitjançant el qual un usuari registrat podrà informar a l'ajuntament que aquella ruta o bé punt de control li agraden. En aquesta acció es mira si l'usuari està registrat. Si ho està, es fa una petició al `LikesServlet` amb l'id d'usuari, del punt/ruta i el tipus de lloc i llavors s'actualitza el botó en la interfície d'acord a l'estat actual (m'agrada/no m'agrada). En cas de que l'usuari no està registrat, se'l notifica de que és una funcionalitat per a usuaris registrats i se'l convida a anar a la pantalla de preferències i omplir el formulari.

Finalment, manquen per implementar, per una part, en el detall de la ruta el botó de preferits i la gestió del botó de suggerir ruta i d'enviar un repte a un amic, i per l'altra part en el detall del punt, mostrar la imatge del tag NFC i la vista panoràmica.

Quan es prem el botó de preferits, es desa la ruta en memòria, per tal de que quan l'usuari filtri per rutes preferides al llistat, aquella ruta estigui disponible com a una de les seves preferides. Els botons de suggerència i de repte, de moment no fan res, ja que mostraran el diàlegs dels amics un cop s'integri facebook amb l'aplicació.

En el detall del punt, es prepara un sistema de recuperació de les imatges, perquè un cop s'hagin descarregat del servei ImageDownloadServlet, es desin a la memòria del dispositiu i no s'hagin de tornar a descarregar.

Per la part de les imatges panoràmiques, inicialment s'utilitza una llibreria que va fer Google que es diu *PanoramaGL*, que manté la compatibilitat de les fotos panoràmiques per a dispositius de la 2.1 en endavant. Degut a les limitacions de la llibreria, restriccions i la qualitat d'imatge obtinguda, s'acaba deixant d'utilitzar la llibreria i es fa que aquesta funcionalitat es faci amb la galeria d'imatges nativa d'android i per tant la vista panoràmica estarà disponible a partir de la versió 4.1 per a la versió d'aquest projecte i més endavant quan la llibreria millori, es tornarà a incorporar per al producte final.

iii) Aquesta funcionalitat realment té un temps de 30 minuts, i la resta es dedicarà a test global d'inconsistència de dades i comprovació de que l'acoblament es fa correctament i no deixen de treballar correctament les funcionalitats anteriors que depenen d'aquest sprint.

L'acoblament entre les pantalles consisteix en integrar tot el que ja havíem fet i les noves pantalles d'aquest sprint, fent el pas d'informació corresponent a cadascuna de les pantalles. Quan es fa clic a una de les rutes que es mostra en el llistat, es passa al detall i cal saber quina ruta s'ha seleccionat, per tal de mostrar les seves dades. El mateix passa en el cas dels punts de control que es seleccionen des de la pantalla del detall.

### **c) Test i valoració d'sprint**

**Test:** Comprovar que no hi ha inconsistència de dades, crear rutes amb molts punts per testear si es pinten bé tots els punts i si es veuen correctament a la pantalla, a més de la reacció d'aquests mapes als gestos que fa l'usuari (zoom/ rotació horitzontal/ angle de vista vertical).

**Valoració:** El temps esperat s'ha assolit de manera satisfactòria, però el temps de les tasques s'ha reduït en 3 hores a les pantalles de detall i s'ha augmentat en aquest temps en temps de documentació i test dels mapes.

### **6.2.3. Sprint 3**

Fins ara les dades s'havien carregat a través de la classe estàtica de test. En aquesta part, es crearà tota la part client (front-end) que cridarà els serveis i desarà les dades en els beans generats en l'sprint 1. A més a més, s'incorpora la part de gamificació a través de Facebook, per tal de que un usuari registrat (el registre o pantalla de preferències

s'implementarà en el següent sprint) pugui enviar i rebre reptes, suggerir una ruta a un amic i dir que aquella ruta o punt de control li agrada (això facilitarà a l'ajuntament saber per quines zones la gent practica més esport i quins tipus de rutes són els que agraden més).

### **a) Planificació**

- i) Integració amb l'API de Facebook per a la part de gamificació. 36 hores
- ii) Diàleg amics. 4 hores
- iii) Acoblament entre l'aplicació i el servidor per a l'obtenció/enviament de dades. 40 hores

### **b) Implementació**

i) El procediment per a integrar Facebook dins de l'aplicació és molt semblant al dels mapes:

- Primerament, abans de res, cal obtenir una API key única per a la nostra aplicació que afegirem al manifest.

- Després, quan ja tenim aquesta clau, ja es pot afegir la llibreria i començar a afegir la funcionalitat, però abans caldrà realitzar l'estudi de la documentació per a obtenir la funcionalitat que es vol a la nostra aplicació.

- Els serveis que necessitem són els següents: fer el log in d'un usuari, obtenir els amics d'aquest i finalment enviar un missatge a un amic per a fer-li un suggeriment de ruta. Per a utilitzar els serveis de l'API és necessari primer de tot buscar quins permisos es necessiten per a fer aquestes accions, ja que el primer cop que l'usuari faci *log in*, se l'avisarà de tot allò que pot obtenir l'aplicació del seu compte. En aquest cas els permisos són *friends\_about\_me* i *publish\_stream*, que serviran per obtenir els amics i publicar en el mur d'aquests.

- Per a fer el log in és necessari utilitzar la classe Facebook que cerca si l'usuari té instal·lada l'aplicació i es fa una autenticació amb *Oauth 2.0* (protocol segur d'autenticació) obtenint un token únic pel dispositiu i que permetrà realitzar accions sense haver de tornar a escriure l'usuari i la contrassenya cada vegada.

- Quan hem aconseguit obtenir els amics de l'usuari, s'estudia la manera d'obtenir només els amics que tinguin instal·lada l'aplicació, que seran els usuaris als quals podrem enviar un repte de temps sobre una ruta concreta. L'objecte *user*, té un atribut que és el de *installed* i que ens informa si aquell usuari està utilitzant l'aplicació des de la qual estem fent la crida.

Ara que ja ho tenim tot gestionat, es prepara el missatge de suggeriment de ruta i ja hem aconseguit tenir la funcionalitat que es cridarà després en els següents casos:

- En la pantalla de preferències es fa el log in.



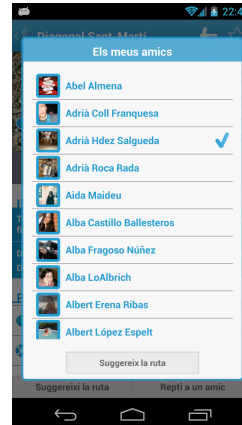
- En el detall de ruta tenim el botó de suggerir ruta, que un cop clicat, obrirà un diàleg amb els amics i també es mostra el botó d'enviar un repte a un amic que tingui instal·lada l'aplicació, que també apareix en la pantalla del resum de pràctica, i que obrirà un diàleg mostrant-ne el llistat. Aquests diàlegs s'implementen a continuació en la següent tasca d'aquest sprint.

ii) Un cop hem integrat Facebook amb l'aplicació i podem obtenir tant els amics d'un usuari en aquesta xarxa social, com els amics que són dins d'ella i a més tenen instal·lada l'aplicació, podem passar a crear el diàleg dels amics. Aquest diàleg ha de permetre acceptar o cancel·lar l'enviament del missatge i seleccionar un dels amics que apareixerà en el llistat (*listview*).

S'implementa un adapter que rep una llista de tots els amics que hem obtingut prèviament i que genera el llistat de cadascun d'aquests amics amb el nom i la foto de perfil. Les fotografies dels usuaris es desen al dispositiu un cop s'han descarregat, així quan es torni a obrir el diàleg, si el nom de la imatge correspon amb una imatge prèviament descarregada, es carrega sense descarregar-la i la vista es carrega més ràpidament.

Si l'usuari selecciona un usuari i prem acceptar, s'envia el missatge de suggeriment de ruta o s'envia el repte a aquest amic. En cas de que premi el botó enrera es tanca el diàleg. El resultat final el podem veure a la següent il·lustració (il·lustració 23):

*Il·lustració 23: Diàleg d'amics*



iii) Fins aquest moment, totes les dades s'obtenien d'una classe de test estàtica basada en el patró Singleton, en la qual teníem mètodes de tipus getter que ens retornaven els objectes amb les dades que volíem mostrar. En aquest moment es decideix incorporar l'acoblament amb el servidor per a mostrar ja les dades reals i dinàmiques que es vagin afegint i eliminant en la base de dades.

Abans de res, eliminem la classe SingletonTest, així veiem de cop els errors en totes les classes en l'aplicació que utilitzaven aquesta classe i no ens oblidarem de preparar cap de les crides que es fan al servidor per a obtenir les diferents dades.

Amb la finalitat de recuperar les dades del servidor, haurem de fer les diferents crides respectives als diferents Servlets esmentats en l'apartat de servidor que ja hem vist anteriorment. Totes aquestes crides s'hauran de fer de manera asíncrona, per tal de no col·lapsar el thread principal de la interfície. Per a assolir això, s'utilitza la classe *AsyncTask* d'Android, que és una classe que a més d'executar-se en un *thread* apart (de tipus asíncron), permet actualitzar els resultats en la interfície un cop s'han descarregat les dades. Aquesta actualització es fa en el mètode *onPostExecute* que és l'últim mètode que es crida en l'*AsyncTask*. Abans d'executar aquestes tasques, es comprova que el dispositiu disposa de connexió a internet en aquell moment per accedir al servidor i si no ho fa, se'l notifica amb un *Toast*, que és un missatge de notificació proporcionat per l'SDK que desapareix al cap d'uns segons.

Cal fer una fase de test intensa, per a comprovar que l'aplicació no deixa de funcionar si tanquem l'aplicació quan descarreguem dades i que les vistes es refresquen correctament un cop es reben aquestes.

### **c) Test i valoració d'sprint**

**Test:** El test d'aquesta fase es centra bàsicament en les proves de consistència de dades quan aquestes es descarreguen del servidor i es volen mostrar, errors de timeout de connexió (temps d'espera esgotat), i control quan s'intenta descarregar i el dispositiu no disposa de connexió o bé el servidor o la *DB* (database) no estan disponibles, casos en els quals s'haurà de notificar a l'usuari.

**Valoració:** La planificació s'ha assolit correctament i a més han sobrat dues hores que han servit per avançar feina per al següent sprint: començar a buscar documentació de l'ús de l'NFC amb els dispositius que tenen com el sistema operatiu Android.

## **6.2.4. Sprint 4**

En aquest sprint s'implementarà la pantalla de preferències, el filtre de la pantalla de l'històric i tota la navegació a través de l'ús de NFC prèvia a la part del tracking.

### **a) Planificació**

- i) Pantalla de preferències (formulari de registre d'usuari). 34 hores
- ii) Filtre per data d'inici i data final a la pantalla de l'històric de pràctiques. 4 hores
- iii) Estudi de l'ús d'NFC a través del dispositiu: restriccions i navegació en l'aplicació depenent del contingut de dades desat en l'etiqueta NFC. 42 hores

### **b) Implementació**

- i) La pantalla de preferències és un formulari d'inscripció d'usuari i conté elements del tipus *button* (botó), *EditText* (espais on l'usuari pot escriure) i *CheckBox* (requadre buit en

cas de que no estigui seleccionat o amb un tick en cas contrari).

El botó de log in es mostrarà en cas de que l'usuari encara no hagi entrat amb el seu compte de Facebook, un cop ja sigui dins, el botó es mou a la dreta descrivint un *log out* (sortir de Facebook) i a l'esquerra mostra el nom d'aquest usuari.

Un cop l'usuari ha emplenat totes les dades i prem el botó desar, es fa una crida al *FormServlet* passant totes les dades per paràmetre i s'afegeix o s'edita l'usuari en cas de que ja existís o no.

Si l'usuari prem el botó i s'ha deixat dades per emplenar, se li notifica amb un missatge en un diàleg, de la mateixa manera que es fa si enlloc de prémer el botó intenta tirar enrera i hi ha camps buits i altres que no ho estan. En aquest cas, si l'usuari ja existia i persisteix en continuar, s'esborrarà l'usuari de la base de dades.

ii) Els filtres de data que apareixen a la part superior de la pantalla de l'històric tenen certa complexitat. Es tracta de dos botons que alhora mostren la data seleccionada per l'usuari i per la qual un cop escollides i es premi el botó d'acceptar (botó OK) es filtraran totes les pràctiques que l'usuari hagi practicat en aquell interval de temps. Internament el codi funciona de la següent manera: s'obté el llistat de totes les pràctiques, s'ordena de manera ascendent i llavors es mira una a una totes les pràctiques per saber si pertanyen a aquell interval de temps. Les que ho fan s'afegeixen en una nova llista de pràctiques que es mostrarà un cop acabi el filtratge i es refresqui la pantalla.

Per a fer la tria d'ambdues dates, es creen dos botons que mostren les dades escollides i un cop clicats, accedeixen a un diàleg amb un calendari. Per a la vista del calendari, s'utilitza una llibreria *CalendarView* que podem trobar a GitHub, desenvolupada per SimonVT. Un cop s'integra la llibreria es modifica l'estil de la vista d'acord al disseny i es mostra en un diàleg on es permet escollir el dia. Aquest diàleg, conté una interfície que avisarà l'activitat de la data seleccionada un cop es premi el botó acceptar, i quan l'activitat rebi la data, modifica la vista del botó mostrant la data escollida. A continuació, es mostren els resultats de la vista del calendari (il·lustració 24):

*Il·lustració 24: Diàleg del calendari*



iii) La falta de coneixements de l'estudiant envers de la tecnologia NFC, es converteix en una àmplia fase de documentació sobre el funcionament, restriccions i diferents tests i exemples previs abans de començar a pensar amb la integració en l'aplicació. En la bibliografia es poden consultar alguns dels enllaços seguits per a aprendre el funcionament, així com petits exemples d'iniciació a la tecnologia.

Es descobreix que quan el dispositiu rep l'event de NFC perquè ha llegit alguna etiqueta, es llegeix el *mime-type* del contingut per tal d'enviar la informació a l'aplicació que correspongui en el mateix dispositiu. Aquest *mime-type* s'haurà desat en l'etiqueta juntament amb la informació, durant el procés d'escriptura previ per a afegir la informació que es llegirà després als dispositius. En el nostre cas s'utilitza el següent mime-type: "application/vnd.tempos21.tapandgo". Després, es llegeix la resta d'informació que conté el tag, és a dir, l'id de la ruta i la posició del punt de control.

Tota aquesta informació s'enviarà a l'activitat NfcDispatcherActivity, on hi ha registrats tots els NfcHandlers que utilitza l'aplicació. Depenent del tipus d'informació que conté el tag, s'enviarà la informació a un handler o a un altre i aquest avisarà a la seva activitat perquè refresqui la vista corresponent que resulta modificada per la nova informació obtinguda. Es fa necessari utilitzar aquest dispatcher, ja que hi ha diverses pantalles que es veuen modificades depenent de la informació i de l'estat actual en l'aplicació i cal gestionar què es farà en cadascun dels casos. Els diferents *handlers* es prepararan en l'sprint que gestioni el tracking d'una pràctica.

### **c) Test i valoració d'sprint**

**Test:** El test consistirà en comprovar que les dades de l'usuari es desen correctament al servidor, ja que en cas de que això no passi, l'usuari no podrà utilitzar la part de gamificació de l'aplicació. És important doncs controlar tot el cicle i re-edició d'aquest formulari, ja que les pantalles que fan ús de la gamificació no es podran utilitzar si la gestió no és correcte. A més a més cal fer un anàlisi exhaustiu dels events NFC i comprovar que la resta de funcionalitats i vistes no es modifiquen quan el HW del dispositiu s'activa en llegir una etiqueta.

**Valoració:** Ha estat necessari dur a terme un estudi a fons de l'NFC, ja que com hem vist, depenent de la informació de l'etiqueta, la interacció és diferent. La planificació s'ha seguit sense problemes, ja que s'estimava un gran nombre d'hores d'estudi i s'havien afegit hores addicionals pels problemes que poguessin sorgir en el desenvolupament d'aquesta funcionalitat.

### **6.2.5. Sprint 5**

Abans d'implementar la part de tracking del projecte, ara es crearà tota la interfície de les pantalles que l'utilitzen i es prepararà i s'optimitzarà de manera que un cop s'afegeixi la lògica del seguiment de la pràctica, els càlculs constants no facin que l'aplicació es "pengi"

o deixi de funcionar a causa de processar masses dades en poc temps. A més a més, Facebook ha fet canvis en el graph de la seva API i el diàleg de suggeriment de la pràctica, ha deixat de funcionar i s'ha de tornar a implementar.

#### **a) Planificació**

- i) Degut als canvis de Facebook en el graph de la seva API: modificació del procés i navegació a l'hora d'enviar un missatge de suggeriment de ruta a un amic. 4 hores
- ii) Creació de les següents pantalles: pantalla prèvia a l'inici d'una pràctica, pantalla que es mostra durant la pràctica de l'esport i finalment la pantalla on es mostren els resultats obtinguts. Només es creen les vistes i es controla el fluxe d'informació entre els events d'NFC i el dispositiu, el tracking es realitzarà en el següent sprint. 75 hores
- iii) Acoblament entre l'aplicació prèvia a aquest sprint i la nova funcionalitat. 1 hores

#### **b) Implementació**

- i) Facebook està fent canvis en la seva API i està començant a treure i modificar funcionalitat. Tal i com anuncia l'enllaç que veurem a continuació la petició d'enviar un missatge de manera transparent ha deixat de funcionar a data del 6 de Febrer.

Web: <https://developers.facebook.com/roadmap/completed-changes/>

Ara ja no es pot utilitzar un diàleg personalitzat i enviar únicament el missatge que es vol, sinó que l'usuari també pot escriure, ja que és obligatori utilitzar el diàleg natiu d'enviament d'un missatge que proporciona Facebook. Encara que això sigui problemàtic ja que l'usuari podrà afegir informació, s'han de fer els canvis i aconseguir que l'aplicació funcioni de manera correcta amb els nous canvis.

- ii) La pantalla prèvia a l'inici d'una pràctica permet seleccionar l'esport que es vol practicar (en cas de que es puguin practicar ambdós), un cop l'usuari ha escollit la ruta. S'informa a l'usuari de que està a punt de començar, es mostra novament el mapa perquè pugui consultar per últim cop on es troben cadascun dels punts i finalment s'incorpora els botons selectors de l'esport i el botó de començar.

Quan ja s'ha seleccionat l'esport, aquest es desa a les dades inicials de la pràctica, així com la ruta i es mostra la pantalla de tracking amb un diàleg informant a l'usuari de que validi el primer punt per a començar. Quan el dispositiu rep l'event de check in en aquest primer punt, s'engegarà el cronòmetre i el servei de tracking que s'implementarà en el següent sprint.

Tal i com s'havia esmentat en la tasca de l'estudi de NFC en l'Sprint anterior, s'implementa un handler que un cop rebut un event de NFC al sistema i s'ha llegit la informació continguda en l'etiqueta, enviarà la informació a una activitat o a una altra i permetrà diferenciar els 3 tipus de punts de control:

- Punt inicial: Tindrem un punt inicial quan es llegeix de l'etiqueta l'id de la ruta i es llegeix un zero com a valor de posició del punt a la ruta.
- Punt intermig: Tindrem un punt intermig quan es llegeix de l'etiqueta l'id de la ruta i el valor de posició no correspon ni a un 0 ni a  $n-1$ , on  $n$  és el nombre de punts de control que té la ruta.
- Punt final: Tindrem un punt final quan es llegeix de l'etiqueta l'id de la ruta i el valor de posició es correspon amb el nombre de punts que té la ruta menys 1 (ja que el primer punt té el valor 0).

Cal tenir en compte que només es duu a terme el control d'aquests events de NFC i es prepara el codi amb la finalitat de que al següent sprint s'afegeixi tota la lògica.

iii) Donada la nova funcionalitat d'aquest sprint, manca acoblar l'aplicació que tenim fins ara amb les noves pantalles. La pantalla de tracking s'assoleix prement el botó començar en el detall de la ruta (botó de sota del mapa). En aquest moment, es direccionarà a la pantalla d'inici de la pràctica (pantalla de selecció de l'esport abans de començar). A més a més, també podrem arribar a aquesta pantalla directament quan es llegeix un tag NFC d'inici de ruta, i fins i tot això funcionarà encara que l'usuari no tingui oberta l'aplicació (l'únic requeriment és tenir l'aplicació instal·lada).

### **c) Test i valoració d'sprint**

**Test:** S'ha testejat de nou la gamificació a través de Facebook, i tests d'interfícies, per a comprovar que el cicle de vida de les activitats es seguia correctament i l'aplicació no deixava de respondre. A conseqüència dels events de NFC, es redibuixaven algunes vistes com les dels mapes. Això s'ha resolt fent un *handler* que s'encarregava només de controlar els events de NFC avisant als components corresponents perquè posteriorment facin l'acció que els pertoca.

**Valoració:** La planificació finalment s'ha seguit, tot i que semblava que això no passaria a conseqüència de les variacions que s'han hagut de fer als mapes que es tornaven a dibuixar quan es rebia l'event de NFC. Gràcies a que la interfície s'ha acabat desenvolupant abans del temps previst, aquest temps addicional no ha influït en la data final.

### **6.2.6. Sprint 6**

Finalment i per acabar, és necessari dur a terme tot el control del tracking d'una pràctica. El contingut d'aquest sprint és precisament aquest, així com la creació del vídeo que es posa a l'ajuda (help) de l'aplicació per tal de veure com funciona l'aplicació de manera ràpida i així l'usuari no tindrà cap problema el primer cop que l'utilitzi. Aquest sprint és diferent als altres ja que conté una gran part de test per a obtenir el menor marge d'error en les dades i per tant durant diversos dies l'estudiant anirà a córrer i s'anirà perfilant

aquesta exactitud per tal de que el temps de gravació del vídeo sigui mínim i no depengui dels errors de precisió del HW.

### **a) Planificació**

- i) Seguiment de la pràctica (tracking): localització, cronòmetre i càlcul de velocitats. 66 hores
- ii) Mostrar les dades reals del tracking en la pantalla de resultats. 4 hores
- iii) Desat de les dades del seguiment al dispositiu. Si l'usuari està registrat, les dades es pujaran al servidor i es controla si ha superat algun dels events de gamificació i, si és així, indicar-ho a l'usuari i actualitzar la base de dades. 4 hores
- iv) Gravació i edició del vídeo de demostració que s'integra a la pantalla d'ajuda (help) de l'aplicació. 6 hores

### **b) Implementació**

i) El servei de tracking de la pràctica és una de les parts més complexes d'aquest projecte. S'han de calcular moltes dades en un interval molt petit de temps i alhora actualitzar la interfície. A més a més s'afegeix l'event de *check-in* quan l'usuari vol validar un punt de control, i per tant addicionalment tindrem quan el sistema rebí un event d'aquest tipus, serà necessari llegir les dades de l'etiqueta i operar el més ràpid possible per no carregar encara més el processament.

Durant aquesta implementació es fa necessari que tot el processament de càlculs i que el cronòmetre s'executin en background en un servei d'Android (*service*), ja que d'aquesta manera millorarem la consistència de dades. El *garbage collector* (alliberador de memòria dinàmica del sistema operatiu) allibera primer els recursos de prioritat baixa i mitja, mentre que els services són tasques de prioritat alta i per tant incrementem el nivell de certesa de que aquests càlculs es seguiran fent, malgrat s'alliberin altres recursos del sistema operatiu. L'ús d'aquest tipus de servei permet també utilitzar l'aplicació amb el dispositiu bloquejat i els càlculs es seguiran realitzant, fins el punt que l'usuari vulgui validar un punt amb NFC, llavors és imprescindible desbloquejar el dispositiu perquè es rebí l'event del HW (event de NFC).

El cronòmetre és un *thread* apart, s'utilitza la classe nativa de java *TimerTask*, i aquest correrà dins del servei esmentat anteriorment comptant cada segon, independentment dels càlculs que es vagin fent i l'actualització de la interfície, així com el desat de les dades dels punts de control. Un cop la pràctica finalitzi o bé l'usuari decideixi finalitzar la pràctica prement el botó enrera, es parará.

La latitud i longitud de la localització de l'usuari en cada moment, s'obindrà a través del mètode `getMyLocation()` del mapa i es tornarà a pintar la vista de l'esportista al mapa cada 3 segons, de la mateixa manera que l'interval de refresc per pantalla de la velocitat a la

qual va l'usuari. Aquesta velocitat s'aconsegueix mitjançant el mètode `getSpeed()` de la localització (*Location*) de l'usuari. És important saber que aquesta localització serà molt més fiable si l'usuari té activat el gps, ja que en cas contrari aquesta s'obtindrà a través de la xarxa del dispositiu mòbil i el marge d'error serà molt més gran.

Finalment, manca preparar el tractament de l'event NFC: que succeeix en la interfície i en el desat de dades en el dispositiu, en cadascun dels punts segons la seva tipologia (punt inicial/ punt intermig/ punt final). Els successos depenent del tipus de punt són els següents:

- Punt inicial: Quan succeeix aquest event, s'engegarà el cronòmetre i el servei que fa tots els càlculs també comença a funcionar.
- Punt intermig: Un cop s'ha validat un punt intermig, cal calcular i desar el resultat en el dispositiu del parcial obtingut.
- Punt final: En aquest cas es desarà l'últim parcial de la pràctica i seguidament s'anirà a la pantalla de resultats per tal de mostrar-los i pujar-los al servidor (en cas de que l'usuari ho vulgui).

ii) Degut a que les dades s'actualitzen en un temps de refresc determinat, i que els càlculs es fan en un servei en *background* en un *thread* diferent al de la interfície, es fa necessari implementar un *handler*, que és un objecte que quan sigui cridat pel thread del servei, notificarà i farà el pas de dades al UI thread perquè s'actualitzi la interfície. Així s'aconsegueix que les actualitzacions siguin assíncrones i per tant només el servei que va fent els càlculs serà el que necessitarà processament constant durant tota la pràctica. Però encara que la resta d'events només necessitaran processament en moments puntuals, cal optimitzar per a minimitzar el temps de processament per tal de que l'aplicació no es pengi fent ús de masses recursos. Per tant, és essencial fer un test exhaustiu.

iii) Un cop l'usuari acaba la pràctica i es mostra la pantalla del resum del tracking, es desa l'informació a la memòria interna del dispositiu. Per a aconseguir això, és deserialitza un fitxer que conté totes les pràctiques que ha fet l'usuari, s'afegeix la nova pràctica i finalment es torna a seralitzar i es desa el fitxer en les preferències d'Android. Si l'usuari està registrat i vol enviar la pràctica al servidor i participar en els events de gamificació disponibles sobre aquella pràctica, ho podrà fer prement el botó "Comparteixi resultats". Quan es fa clic en aquest botó, s'envia la informació a través del servei de pràctiques (servlet) passant tota la informació que aquest necessita. Llavors el servei puja la pràctica al servidor i comprova si s'ha superat algun dels events d'aquesta ruta, retornant al client (aplicació android) un array amb els id's dels events superats. L'aplicació en rebre aquesta informació, mostra en un diàleg els events que l'usuari ha superat (si és que n'hi ha algun).



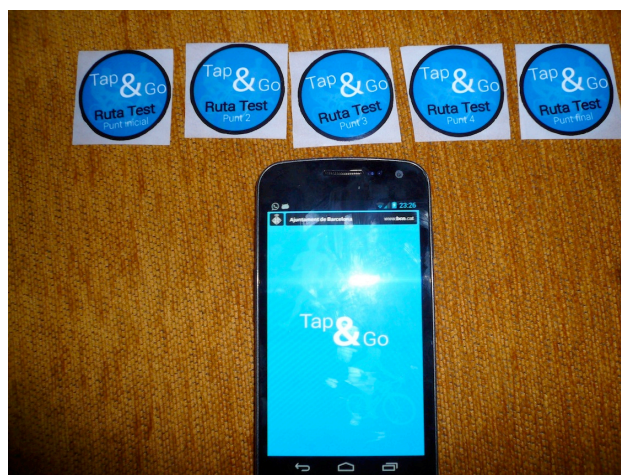
iv) Es prepara una ruta, els seus punts i les etiquetes NFC que s'enganxaran pels punts de control. S'afegeix l'informació de la ruta i dels punts en la base de dades i després es desa l'informació corresponent en nous *tags* a través de l'aplicació de gestió, que també ens servirà per llançar les fotografies de la localització de les etiquetes un cop estiguin enganxades per la població de Parets del Vallès (és on l'estudiant realitza les proves i aquest vídeo).

Abans de gravar és essencial comprovar que tots els paràmetres i els marges d'error en el tracking són mínims, per tal de que abans de res, es comprovi que tot funciona correctament. A conseqüència dels problemes amb el gps sorgits en la fase de test, la data de gravació d'aquest vídeo s'ha anat allargant, alhora que de manera directa s'ha hagut de tornar a entrar en la primera part (primera tasca) d'aquest sprint, modificacions en el tracking.

Un cop tot ha funcionat correctament i s'ha obtingut el vídeo final, aquest s'edita amb iMovie i es prepara per a convertir-lo en el vídeo final que s'integrarà al help de l'aplicació i que permetrà veure als ciutadans com fer ús de l'aplicació en una pràctica i validant els punts amb L'NFC del dispositiu.

En la següent il·lustració (il·lustració 25) podem veure l'equip emprat en el testing.

*Il·lustració 25: Dispositiu de test i etiquetes NFC*



### **c) Test i valoració d'sprint**

**Test:** Aquest test ha estat el més complicat de tots ja que ha estat realment llarg i problemàtic. Degut a la localització i càlculs de velocitats, l'sprint s'ha allargat, ja que el marge d'error era bastant gran perquè les velocitats es calculen emprant la fórmula física ( $\text{velocitat} = \text{distància} / \text{temps}$ ) i la distància obtinguda mitjançant el gps del dispositiu era errònea. Ampliant el temps de l'sprint, a poc a poc s'ha anat millorant el marge d'error.

**Valoració:** La valoració d'aquest sprint no és positiva en la planificació, ja que l'sprint ha acabat durant dues setmanes més de les previstes a conseqüència dels problemes sorgits durant el test i gravació del vídeo. Malgrat això, els resultats obtinguts finalment es poden

considerar prou precisos, tenint en compte però que s'ha hagut de canviar el lloc de test, ja que en la ruta prevista inicialment el gps tenia problemes importants en la localització.

### **6.2.7. Sprint 7**

Un cop desenvolupada l'aplicació dels ciutadans, cal implementar la de gestió. Es deixa pel final ja que el contingut d'aquesta és molt menys extens i es pot aprofitar molt de codi de l'altra aplicació. En l'apartat de disseny hem pogut comprovar que aquesta aplicació només té la pantalla del llistat de rutes, la del detall d'una ruta i finalment la d'un punt dins d'una ruta, i que a més a més, aquestes tenen un comportament diferent al de l'aplicació dels ciutadans. Addicionalment, a l'inici de l'aplicació trobem un *log in* per tal que l'usuari de tipus administrador validi les seves credencials.

#### **a) Planificació**

- i) Còpia de l'aplicació dels ciutadans i adaptació a l'aplicació de gestió. Aquesta part inclou eliminar totes les dades que no s'utilitzen en aquesta aplicació i modificar les parts on hi ha diferència respecte l'altra aplicació. 6 hores
- ii) Pantalla de log in. 6 hores
- iii) Pantalla d'edició d'un punt de control. 68 hores

#### **b) Implementació**

i) Primerament es fa una còpia de l'aplicació dels ciutadans i es treuen totes les pantalles i funcionalitats que no corresponguin a aquesta aplicació. D'entrada doncs, un cop s'ha creat la còpia, s'eliminen la pantalla de l'històric, la dels amics de facebook que tenen instal·lada l'aplicació, la pantalla de premis, de preferències i finalment la d'ajuda, així com tots els accessos directes (icones) que permetien accedir a aquestes pantalles, com per exemple el botó de començar pràctica (en el detall de la ruta) i els botons de gamificació.

El menú superior tampoc serà necessari, per tant també l'eliminem juntament amb les icones de preferits, m'agrada, accés directe a la pantalla de reptes i premis, etc.

Després venen els canvis interns en la pantalla de rutes: es canvia el filtre de preferits per un filtre de rutes no visibles als ciutadans. Aquestes rutes són totes aquelles que els usuaris administradors estan preparant i per tant, no interessa mostrar als usuaris. Aquestes s'identifiquen a través del camp de *visibility* a 0 en la taula de rutes de la base de dades i a diferència de l'aplicació dels ciutadans, també ens interessa controlar-les ja que permetrà a l'administrador saber de manera ràpida el què és visible i el què no. Per a aconseguir aquest comportament en el servei de descàrrega de rutes obtindrem totes les rutes (visibles i no visibles) i mitjançant el filtre de visibilitat s'aconseguirà veure quines són totes aquelles rutes que poden veure els ciutadans i quines no perquè estan en procés d'edició.

ii) La pantalla de log in permet validar les credencials d'un usuari de l'aplicació d'administració. D'aquesta manera serà senzill saber qui ha realitzat els canvis en els punts de control en cadascun dels moments i s'assegura que encara que l'aplicació només la tindrà el personal de l'ajuntament, tampoc es pugui accedir si no es tenen les credencials. Així s'evita que només pugui editar contingut el personal autoritzat.

Un cop l'administrador escriu el nom d'usuari i contrassenya en els *EditText* (formularis de text), quan es prem el botó d'acceptar es crida al servei de log in passant com a paràmetre aquestes dades i un cop comprovades i són vàlides es mostra l'activitat (pantalla) del llistat de rutes. En cas de que són incorrectes, es mostra una notificació per pantalla.

iii) Finalment, es convertirà la pantalla del detall d'un punt de control en una pantalla d'edició on es podrà assignar aquest punt a una etiqueta NFC des del dispositiu i també es podran pujar una fotografia de la localització d'aquest tag (etiqueta) i una altra de panoràmica. Es prepararà per poder editar aquests elements un cop ja s'hagin assignat, per tal de si es volen canviar, hi hagi un mode d'edició d'aquesta informació.

En l'etiqueta NFC s'escriu l'identificador de la ruta i la posició d'aquest punt de control dins d'aquesta i seguidament quan l'administrador prem el botó de desar canvis es desa l'identificador únic de l'etiqueta NFC (cada etiqueta NFC del món té un id únic format per caràcters i nombres), és a dir, s'actualitza aquest id d'etiqueta en la taula del punt de control de la base de dades mitjançant una crida al servei PointServlet.

Seguidament, es treballa la funcionalitat per a poder pujar la fotografia des del dispositiu. Comentar que la finalitat d'aquestes fotografies és que un cop s'enganxi l'etiqueta al carrer, faciliti a l'usuari saber-ne la seva localització, ja que a través del mapa podem identificar el lloc on està col·locada a través d'un punt, però no es pot veure el paisatge d'ubicació real. Per això, es pot afegir a més a més una fotografia panoràmica que permet veure a més a més el voltant des de davant de l'etiqueta fins a una rotació de 360 graus.

### **c) Test i valoració d'sprint**

**Test:** Es fan proves per a validar que si hi ha algun problema pujant informació (imatges i dades) al servidor, l'aplicació no deixi de funcionar. També es comprova que les dades que es pugen són correctes i que des de l'aplicació dels ciutadans es mostra tot de manera correcta, especialment les imatges, que per problemes de codificació (*encoding*), al principi no es mostraven correctament al dispositiu.

**Valoració:** Malgrat l'sprint semblava tenir una complexitat baixa, la funcionalitat de pujar imatges des del dispositiu al servidor ha donat bastants problemes. Ha estat necessari mirar molta informació en diverses webs i fer diverses modificacions en el servei de pujar imatges al servidor, per tal d'aconseguir que aquest funcionés correctament.

No obstant, el dia de finalització de l'sprint no ha variat, ja que les altres tasques s'han

finalitzat abans del temps previst i s'han afegit 6 hores addicionals repartides entre els 2 últims dies per tal d'assolir la data.

## 7. Conclusions

Un cop s'ha arribat a aquest punt i el projecte ja es dona per acabat, cal fer una valoració de si els objectius s'han assolit, l'estat actual en el mercat del producte, així com la valoració extreta del temps d'estudi i implementació del projecte.

### 7.1. Assoliment dels objectius

Analitzant els requeriments o objectius que es van esmentar a l'inici del projecte en l'estudi de viabilitat, ens adonem de que aquests s'han acabat complint tots de manera satisfactòria.

El projecte s'ha acabat endarrerint dues setmanes degut al marge de precisió de les dades, durant el test en la ruta real, ja que el GPS dels dispositius mòbils té un error bastant gran en molts casos. S'ha acabat solucionant incrementant la dependència entre els punts físics (etiquetes NFC) i re-calculant les dades depenent dels trams entre punts.

Tot i intentar dur a terme un bon control de la planificació en l'estudi, es tracta d'un projecte molt gran, on hi ha una gran quantitat d'innovació i per tant aprenentatge, i era un dels riscos que podia passar.

D'altra banda, els objectius relacionats directament amb el posicionament del producte al mercat, encara no s'han pogut comprovar. L'ajuntament, com a client, ha vist el producte i està content, així que s'acabarà de perfilar la solució final implementant els requeriments posteriors al projecte (s'esmenten en l'estudi de viabilitat), així com muntant la base de dades i el Servlet al seu propi servidor, i si fos necessari incrementar la seguretat d'aquesta, ja que la focalització del projecte eren les dues aplicacions mòbils.

L'empresa on l'estudiant ha fet el projecte, troba que el projecte ha acabat desencadenant en un producte amb una gran qualitat i per tant s'obren noves portes de comercialització envers el mateix client (l'ajuntament), però també d'altres, gràcies a l'ús de dues tecnologies: l'NFC i els mapes en 3 dimensions, ja que són tecnologies que estan començant a créixer ara i poden aportar nous projectes amb altres clients.

S'espera que el producte aculli un gran nombre de ciutadans i que l'aplicació es converteixi en un complement més per a tots els esportistes de la ciutat.

### 7.2. Línies futures

Les aplicacions compleixen totalment amb la seva funcionalitat i a més el seu funcionament és bastant òptim. De tota manera, podem indicar com a línies futures: la part del tracking, afegir una nova funcionalitat per a calcular calories consumides en la

pràctica, afegir nous esports, fixar un límit més escaient depenent de cadascun dels trams d'una ruta (depenent de si el tram és de pujada o baixada, perfilar els temps que es poden obtenir depenent del tipus d'esport), etc. Addicionalment a aquests canvis, el disseny anirà canviant d'acord al temps i les novetats en la plataforma d'Android i quan el client ho sol·liciti.

### **7.3. Valoració personal**

Després de la gran implicació en el projecte i haver-lo pogut finalitzar satisfactòriament, la meva valoració és molt positiva.

He adquirit un gran nombre de coneixements, he après sobre les últimes tecnologies i he estat capaç d'afrontar un projecte bastant gran de manera autònoma. També ha estat molt important veure la gran quantitat de documentació i d'estudi que hi ha darrera d'un projecte, ja que quan es treballa en equip, cadascú té el seu rol i molts cops per motius de departaments dins de l'empresa, no es té l'oportunitat de saber quin paper juguen cadascun dels companys.

L'elaboració de la memòria ha estat difícil, ja que s'han hagut de resumir breument 6 mesos de treball, per tant s'ha hagut de fer una bona selecció del contingut, intentant obtenir la millor expressió de la idea del projecte.

També és molt important saber que l'empresa està agraïda i contenta amb el resultat, ja que és aquí on vaig començar a desenvolupar SW per a la plataforma Android, i per tant, on he rebut tota la formació que m'ha permès créixer i haver estat capaç de realitzar aquest projecte, veient un menor grau de complexitat, gràcies als coneixements previs.

Aquest projecte posa fi a una etapa de la meua vida, que ha estat difícil en molts moments, però en la qual he après molt i he pogut acabar decidint què és el que realment m'agrada i, per tant, trobar el punt de partida per a seguir lluitant pel meu futur.

## Agraïments

Vull agrair el suport ofert per l'empresa durant tot aquest temps, per donar-me l'oportunitat de fer un projecte professional d'aquestes característiques, la seva ajuda, així com la seva amabilitat en el servei d'horaris que m'han ofert perquè pogués finalitzar.

També vull donar les gràcies a la meva tutora, que no només ha ofert la seva amabilitat per a dur a terme el control del meu projecte i fer-ho de manera excel·lent, sinó que també ha estat disponible per a resoldre'm dubtes en tot moment.

Finalment a la meva mare i el meu germà que han estat tot aquest temps fent-me costat en els moments més durs, a la família i als amics, que durant tot aquest temps m'han donat forces quan més ho he necessitat. I especialment, al meu pare, de qui vaig aprendre, entre altres coses, que la lluita envers la vida és essencial. Tota la vocació i amor que he avocat dins d'aquest projecte, el dedico a la seva persona.





## Bibliografia

Totes les adreces web i llibres que es mencionaran a continuació s'han emprat durant tot el desenvolupament del projecte, degut a l'iteració per Sprints fent l'ús d'Scrum.

### ADRECES WEB (WEB-GRAFIA)

**Adreça:** <http://developer.android.com/index.html>

**Descripció:** Web oficial per a desenvolupadors de la plataforma Android.

**Llengua:** Anglès

**Adreça:** <http://sourcemaking.com/>

**Descripció:** Web amb informació de disseny de patrons, UML, amb exemples, vídeos, etc.

**Llengua:** Anglès

**Adreça:** <http://www.netmagazine.com/tutorials/getting-started-nfc-android>

**Descripció:** Introducció al desenvolupament d'aplicacions NFC en la plataforma Android.

**Llengua:** Anglès

**Adreça:** <https://www.lucidchart.com/>

**Descripció:** Servei de creació de diagrames UML, sketch, documentació... i guardat d'aquests documents al núvol.

**Llengua:** Anglès

**Adreça:** <http://www.tutorialspoint.com/jdbc/jdbc-transactions.htm>

**Descripció:** Tutorial per a aprendre a utilitzar transaccions en els serveis d'accés a la base de dades.

**Llengua:** Anglès

**Adreça:** <http://stackoverflow.com/>

**Descripció:** Fòrum de resolució de dubtes i problemes en el desenvolupament de solucions SW en diferents plataformes i llenguatges de programació.

**Llengua:** Anglès

**Adreça:** <http://balusc.blogspot.com.es/>

**Descripció:** Tutorial per a aprendre a implementar un servei per pujar i descarregar imatges a un servidor.

**Llengua:** Anglès

**Adreça:** <https://developers.facebook.com/docs/opengraph/>

**Descripció:** Documentació relativa a l'ús dels serveis de l'API de Facebook.

**Llengua:** Anglès

**Adreça:** <https://developers.google.com/maps/documentation/android/>

**Descripció:** Documentació relativa a l'ús de l'última API de Google Maps en Android.

**Llengua:** Anglès

**Adreça:** <http://www.netmagazine.com/tutorials/getting-started-nfc-android>

**Descripció:** Petit tutorial introductori a l'ús de NFC en dispositius Android.

**Llengua:** Anglès

**Adreça:**

<http://caughtinthemobileweb.wordpress.com/2011/06/20/how-to-implement-calendarview-in-android/>

**Descripció:** Algunes de les vistes de calendari disponibles en la plataforma Android.

**Llengua:** Anglès

**Adreça:** <http://www.bcn.cat/trobatb/ca/circuits-esportius.html>

**Descripció:** Web d'esports on es poden veure els circuits esportius disponibles actualment en la ciutat de Barcelona.

**Llengua:** Català

**Adreça:** <http://docs.oracle.com/javase/7/docs/api/index.html>

**Descripció:** Documentació de Oracle de l'API de Java

**Llengua:** Anglès

**Adreça:** [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page)

**Descripció:** Definicions de les abreviacions.

**Llengua:** Anglès

**Adreça:** <http://www.scrum.org/>

**Descripció:** Informació pertanyent a l'Scrum.

**Llengua:** Anglès

## **LLIBRES**

**Títol:** "REST in Practice" Hypermedia and Systems architecture

**Autor:** Jim Webber, Savas Parastatidis, Ian Robinson

**Editorial:** O'Reilly

**Any:** 2010

**Llengua:** Anglès

**Títol:** "Gamification by Design" Implementing game mechanics in web and mobile apps

**Autor:** Gabe Zichermann, Christopher Cunningham

**Editorial:** O'Reilly

**Any:** 2011

**Llengua:** Anglès

**Títol:** Professional Android 4 Application Development  
**Autor:** Reto Meier  
**Editorial:** Wrox  
**Any:** 2012  
**Llengua:** Anglès

L'estudiant, *Jonathan Frago*so Pérez,  
i que ha realitzat el projecte a l'empresa *Tempos 21*  
(*Atos*),

**CERTIFICA:**

Que realitza l'entrega d'aquest projecte  
I per a que consti firma la present.  
Sabadell, *Abril* de *2013*

-----  
Signat: *Jonathan Frago*so Pérez





